2019-5

Phase-Only Digital Encryption

Jonathan Blackledge
Technological University Dublin, jonathan.blackledge@tudublin.ie

Western Govere
University of KwaZuluNatal, Republic South Africa

Dumisani Sibanda
University of KwaZuluNatal, Republic South Africa

Recommended Citation

Blackledge. J.,Govere.W.,& Sibanda, D. "Phase-Only Digital Encryption," *IAENG International Journal of Applied Mathematics,* vol. 49, no. 2, pp212-228, 2019 doi.org/10.21427/jccn-2q79

Phase-Only Digital Encryption

Jonathan Blackledge, Western Govere and Dumisani Sibanda

***Abstract*—We study the n-dimensional deconvolution problem associated with an impulse response function and an (additive) noise function that are both characterised by the same phase-only stochastic spectrum. In this case, it is shown that the deconvolution problem becomes well-posed and has a general solution that is both exact and unique, subject to a re-normalisation condition relating to the scale of the solution. While the phase-only spectral model considered is of limited value in general (in particular, problems arising in the fields of digital signal processing and communications engineering, specifically with regard to the retrieval of information from noise), its application to digital cryptography has potential. One of the reasons for this (as discussed in this paper), is that it provides a method of encrypting data where the diffused plaintext can be effectively embedded in a (phase-only) cipher (subject to the floating point precision used for data processing), thereby fully dissipating the statistical signature of the plaintext in the distribution of the cipher. Further, a decrypt can be generated that is computationally efficient subject to the usual cases of sender and receiver having access to identical algorithm(s) and key(s), deconvolution being equivalent to decryption in the context of the (phase-only) encryption model that is considered. For the two-dimensional case, this approach has a potential weakness in terms of a 'correlation attack' using phase retrieval algorithms and a solution to this problem is provided by introducing a (stochastic) amplitude weighting function. Prototype MATLAB functions are provided in the Appendices that accompany this paper to give readers the opportunity to repeat the computational results presented and extend them further. The functions constitute a symmetric algorithm for encrypting and decrypting full colour images in which the key(s) have been exchanged *a priori*. In this context, the final part of the paper considers the application of phase-only encryption for key exchange using a Three-way Pass Protocol for which a further prototype MATLAB function is provided for validation and further development of the approach by interested readers.**

***Index Terms*—Deconvolution, phase-only spectrum, stochastic phase-only functions, cryptanalysis, image encryption, key exchange algorithm, three-way pass protocol, post-quantum cryptography.**

I. Introduction

THE deconvolution problem (for additive noise) has been studied widely and is known to be an ill-posed problem in general with no exact solutions, thereby requiring the application of regularisation techniques to obtain an approximation, subject to certain conditions and criteria, e.g. [1] and [2]. However, in special cases, exact and unique solutions are available, and, in this paper, we consider the deconvolution problem for the case when both the Impulse Response Function (IRF) and (additive) noise function are characterised by a phase-only spectrum, deriving an exact and unique solution subject only to a re-normalisation condition. While this case is not relevant to general purpose applications in signal and image processing, for example, it does have applications in the field of cryptography, namely, phase-only digital encryption.

Manuscript received September 25, 2018; revised version submitted March, 27, 2019.

J. M. Blackledge is Honorary Professor at the Technological University Dublin, Republic of Ireland, a Distinguished Professor at the Centre for Advanced Studies, Warsaw University of Technology, Poland, Professor Extraordinaire, University of Western Cape, Republic of South Africa and Visiting Professor, University of Wales, UK. e-mail - see https://www.linkedin.com/in/jonathan-blackledge-7643a5150/

W. Govere and D. Sibanda are research students in the School of Mathematics, Statistics and Computer Science, University of KwaZulu Natal, Republic South Africa.

A. Background and Purpose

We consider a method of encryption using a phase-only spectrum characterised by a stochastic phase function. This approach has been studied previously by many authors but the focus of such studies has almost exclusively been on optical information security systems and optical encryption, e.g. [3] and [4]. This includes the use of generalised phase-contrast methods implemented using liquid-crystal spatial light modulators to generate binary phase-encrypted masks and phase-only encryption schemes using phase-encoded exclusive-OR rules in the Fourier plane for single path decryption [5]. Phase-only encryption based on phase-shifting interferometry has also been considered [6] although this approach has recently been shown to have critical security issues using ciphertext-only attacks when the ciphertext(s) can be cracked directly without the keys of the cryptosystem [7]. Thus, while phase-only encryption has and continues to be implemented in optics, judging from the open literature, a generic approach to phase-only digital encryption (which facilitates numerical and programmable concepts which are not feasible in optical cryptography) does not appear to have been studied, and, in this context, to the best of the author's knowledge, the ideas and results presented in this paper represent an original contribution to the field. Further, using a digital approach to phase-only encryption provides a solution to ciphertext-only attacks that can not be readily implemented optically. Such a solution is considered in this paper following a cryptanalysis of the algorithms developed for encrypting full-colour digital images.

B. Structure of Paper

We consider a phase-only digital encryption scheme, presenting the theoretical framework for phase-only deconvolution in n-dimensions. Section II briefly introduces the principles of encryption and the convolutional encoding/encryption model which is the underlying basis for the work reported. Section III introduces the principal theorems (specifically Theorem III.1 and associated Corollary's) upon which the algorithms development is based. The application of Theorem III.1 for phase-only encryption is the subject of Section IV which is coupled with a statistical analysis as considered in Section V. The cryptanalysis associated with phase-only digital encryption is given in Section VI which illustrates the potential weakness of the approach under a phase retrieval

attack for which a solution is proposed. Generic encryption and decryption algorithms are then considered in Section VII with Section VIII providing some example results and a short study based on prototype MATLAB functions for encrypting and decrypting full colour digital images, the MATLAB functions being presented in Appendix B. Section IX then explores the application of phase-only encryption for key exchange using a three-pass protocol. The software development and its use is qualified in Section X with concluding statements, related issues and future directions being presented in Section XI. The overall purpose of this paper is to explore the theoretical background and relative simplicity of implementing digital phase-only encryption, its diversity in regard to numerical implementation and some example applications. Such applications can potentially be coupled to existing encryption and encrypted information hiding schemes including those that may be considered to be associated with post-quantum cryptography as briefly explored in this paper.

II. Encryption Models

The models studied in this paper apply to any dimensions n and we therefore consider a theoretical development for the multi-dimensional case. Thus, for a square integrable n-dimensional function $f(\mathbf{r}) \in L^2(\mathbb{R}^n) : \mathbb{C} \to \mathbb{C}$, we define the Fourier and inverse Fourier transforms in 'non-unitary' form as

$$F(\mathbf{k}) = \mathcal{F}_n[f(\mathbf{r})] \equiv \int_{-\infty}^{\infty} f(\mathbf{r}) \exp(-i\mathbf{k}\cdot\mathbf{r}) d^n\mathbf{r}$$

and

$$f(\mathbf{r}) = \mathcal{F}_n^{-1}[F(\mathbf{k})] \equiv \frac{1}{(2\pi)^n} \int_{-\infty}^{\infty} F(\mathbf{k}) \exp(i\mathbf{k}\cdot\mathbf{r}) d^n\mathbf{k}$$

respectively where $\mathbf{k}$ is the spatial frequency vector and $\mathbf{k}\cdot\mathbf{r} = k_1 r_1 + ... + k_n r_n$. These integral transforms define an (n-dimensional) Fourier transform pair which we write using the notation

$$F(\mathbf{k}) \leftrightarrow f(\mathbf{r})$$

where $r \equiv| \mathbf{r} |$ and $k \equiv| \mathbf{k} |$. Further, we define the n-dimensional delta function as

$$\delta^n(\mathbf{r}) = \mathcal{F}_n^{-1}[1] \equiv \frac{1}{(2\pi)^n} \int_{-\infty}^{\infty} \exp(i\mathbf{k}\cdot\mathbf{r}) d^n\mathbf{k} \qquad (1)$$

and consider a space invariant linear system characterised by the equation

$$s(\mathbf{r}) = p(\mathbf{r}) \otimes f(\mathbf{r}) + n(\mathbf{r}) \equiv \int_{-\infty}^{\infty} p(\mathbf{r}-\mathbf{r}') f(\mathbf{r}') d^n\mathbf{r}' + n(\mathbf{r}) \qquad (2)$$

where $[s(\mathbf{r}), p(\mathbf{r}), f(\mathbf{r}), n(\mathbf{r})] \in L^2(\mathbb{R}^n) : \mathbb{C} \to \mathbb{C}$ and $\otimes$ denotes the n-dimensional convolution integral.

In Equation (2), $s(\mathbf{r})$ is the output of the system (the 'signal'), $p(\mathbf{r})$ is the characteristic IRF of the system (i.e. the output signal obtained when $f(\mathbf{r}) = \delta^n(\mathbf{r})$ and $n(\mathbf{r}) = 0$) associated with the detection of the information function $f(\mathbf{r})$ and $n(\mathbf{r})$ is a stochastic function which denotes the 'noise' associated with a measurement of the signal $s(\mathbf{r})$. The stochastic function $n(\mathbf{r})$ is assumed to have some characteristic Probability Density Function (PDF) denoted by $\Pr[n(\mathbf{r})]$ and may be taken to be a complex multivariate PDF composed from the real and imaginary components of $n(\mathbf{r})$, interpreted in terms of the joint distribution of two real random variables. Unlike $n(\mathbf{r})$, $p(\mathbf{r})$ and $f(\mathbf{r})$ are taken to be deterministic functions. Thus, the signal $s(\mathbf{r})$ is the sum of a deterministic function $p(\mathbf{r}) \otimes f(\mathbf{r})$ and a stochastic function $n(\mathbf{r})$. Equation (2) is the 'standard model' for a multi-dimensional signal and assumes that the system it describes is stationary.

A. Encryption Models

A simple encryption model can be interpreted in terms of the signal being given by the sum of the information function $f(\mathbf{r})$ and noise function $n(\mathbf{r})$, expressed as

$$s(\mathbf{r}) = f(\mathbf{r}) + n(\mathbf{r}) \qquad (3)$$

where all functions are typically, but not exclusively, taken to be real. Equation (3), represents the case when $p(\mathbf{r}) = \delta^n(\mathbf{r})$ in Equation (2). Under 'natural conditions' such as the transmission of $f(\mathbf{r})$ through an additive noisy environment, the noise can not be controlled, only classified; specifically, but not exclusively, in terms of its statistical distribution and moments thereof. However, in the field of cryptography, the noise term is under the control of the cryptographer and Equation (3) can be interpreted through the following terms:

$$\text{Ciphertext} = \text{Plaintext} + \text{Cipher}$$

where the cipher (typically a data stream composed of pseudo-random numbers) is generated by some key dependent algorithm so that we can write

$$Algol(key) \to n(\mathbf{r})$$

Here, the notation $\to$ is taken to mean that the output of $Algol(key)$ is the noise function $n(\mathbf{r})$. In this context, there are two principal problems that are common to all digital encryptions systems, namely, given that Alice and Bob have the same algorithm $Algol(key)$:

- How should Alice and Bob agree upon the key(s) used in the cipher generating algorithm - the 'Key Exchange Problem'?
- How can we make an algorithm produce suitable noise - the 'Cryptographic Strength Problem'?

B. Integer and Binary Encryption

In the context of Equation (3), it is clear that decryption of the ciphertext is trivial and given by

$$f(\mathbf{r}) = s(\mathbf{r}) - n(\mathbf{r})$$

If the functions $s(\mathbf{r}), f(\mathbf{r})$ and $n(\mathbf{r})$ are now taken to be discrete n-dimensional floating point arrays, then we can consider this result to be an example of floating point encryption/decryption where the floating point accuracy is taken to be given by the limited bit representation available. By way of an example, for $\mathbf{r} \in \mathbb{R}^1$, when the functions in question are denoted by the vectors $\mathbf{s}, \mathbf{f}$ and $\mathbf{n}$ where each vector is taken to be composed of an equal number of floating-point elements with a precision of p-bits say, the

floating-point space encryption/decryption process is given by

$$\text{Encryption}: \ \mathbf{s} = \mathbf{f} + \mathbf{n}; \quad \text{Decryption}: \ \mathbf{f} = \mathbf{s} - \mathbf{n}$$

For the case when the vectors $\mathbf{s}, \mathbf{f}$ and $\mathbf{n}$ consist of integer values, the process is of course identical, and, more specifically, for a 7-bit ASCII, the integer space encryption/decryption process is given by

$$\text{Encryption}: \ \mathbf{s} = (\mathbf{f} + \mathbf{n}) \text{mod}(127)$$

$$\text{Decryption}: \ \mathbf{f} = (\mathbf{s} - \mathbf{n}) \text{mod}(127)$$

where mod denotes the standard modulo operation. Further, if we consider the vectors to be the binary streams $\mathbf{s}_b, \mathbf{f}_b$ and $\mathbf{n}_b$, respectively, each consisting of an equal number of bits, then the binary space encryption/decryption process is given by

$$\text{Encryption}: \ \mathbf{s}_b = \mathbf{f}_b \oplus \mathbf{n}_b; \quad \text{Decryption}: \ \mathbf{f}_b = \mathbf{s}_b \oplus \mathbf{n}_b$$

where $\oplus$ is the standard Exclusive OR operator.

C. Floating Point and Analogue Encryption

Compared to integer and binary based encryption/decryption, it is important to appreciate that if Alice and Bob encrypt plaintext using a cipher with a p-bit floating point precision data processor and an attack is attempted using q-bit floating point precision where $q < p$, then, providing the computation of the cipher by Alice and Bob exploits the full 'floating point depth' available with p-bits, a decrypt can not be generated directly by an attacker 'armed' with a q-bit precision processor. This is due to a lack of the floating point precision required to decrypt a p-bit precision ciphertext. This issue naturally leads on to the potential use of analogue computing for generating analogue ciphers which may or otherwise be digitised. Although a detailed account of analogue encryption lies beyond the scope of this paper, it is worth stating that this approach is used in military communications and has potential applications when the key exchange problem is not an issue. Such a scenario occurs when Alice and Bob wish to archive data on the cloud when key exchange algorithms become irrelevant as in this case Alice and Bob can encrypt their data using their own key(s) without the need to exchange them between themselves with any other third party. In this case, the potential exists to exploit specialist embedded analogue devices for exclusive use in localised cloud security [17]. In the interim and/or in parallel, users can develop their own digital floating-point algorithms using, for example, the approach taken by Blackledge et al. [9]in which personalised ciphers can be designed using evolutionary computing for, amongst other applications, encrypting 'data on the cloud'.

D. Diffusion and Confusion

An important aspect in the development of encryption algorithms, at least within the context of a given encryption model, is the condition of maximising the levels of Confusion and Diffusion in any transformation of an (input) plaintext to an (output) ciphertext. Confusion refers to making the correlation between the key and the ciphertext as complex and intricate as possible. Diffusion refers to the property that the redundancy in the statistical distribution of the plaintext is dissipated in the distribution of the ciphertext [18]. Ideally, what is required is a process that outputs a uniformly distributed ciphertext. The process of encrypting data using a specific cipher (with a known distribution) and a specific encryption process, does not always guarantee an output ciphertext that is uniformly distributed. However, what matters most is that the distribution of the plaintext is dissipated effectively over the full extent of the plaintext.

Given Equation (2), we consider the following encryption model which is based on replacing the IRF $p(\mathbf{r})$ with the noise function $n(\mathbf{r})$ leading to the equation

$$s(\mathbf{r}) = \overbrace{\underbrace{n(\mathbf{r}) \quad \otimes \quad f(\mathbf{r})}_{\text{Stochastic Diffusion}} \ + \ n(\mathbf{r})}^{\text{Stochastic Confusion}} \tag{4}$$

This encryption model is the foundation upon which the methods considered in this paper are constructed. It is an approach whereby the signal is taken to be the result of the sum of the noise and the transformation of the information function $f(\mathbf{r})$ by the same noise function $n(\mathbf{r})$, the transformation being based on the convolution operation and thereby representing a form of convolutional encoding. We refer to the convolution operation $n(\mathbf{r}) \otimes f(\mathbf{r})$ as the process of (stochastic) diffusion and the addition of $n(\mathbf{r})$ as the process of (stochastic) confusion as illustrated in Equation (4). In this case, both terms on the RHS of Equation (4) are stochastic functions.

Stochastic diffusion is 'maximised' by ensuring that $n(\mathbf{r})$ is uniformly distributed statistically with a uniformly distributed Power Spectral Density Function (PSDF). Maximum confusion is determined by the extent to which $n(\mathbf{r})$ dominates $s(\mathbf{r})$ which we can express in terms of the condition

$$\|n(\mathbf{r}) \otimes f(\mathbf{r})\|_p << \|n(\mathbf{r})\|_p$$

where the p-norm is defined as

$$\|f(\mathbf{r})\|_p \equiv \left(\int\limits_{\mathbb{R}^n} \mid f(\mathbf{r}) \mid^2 d^n\mathbf{r} \right)^{\frac{1}{p}}, \ 1 \leq p \leq \infty$$

with the uniform norm being defined by

$$\|f(\mathbf{r})\|_\infty = \sup\{\mid f(\mathbf{r}) \mid, \mathbf{r} \in \mathbb{R}^n\}$$

We note that, via the Convolution Theorem, Equation (4) can be written in n-dimensional Fourier space as

$$S(\mathbf{k}) = N(\mathbf{k})[1 + F(\mathbf{k})]$$

where

$$S(\mathbf{k}) = \mathcal{F}_n[s(\mathbf{r})], \ F(\mathbf{k}) = \mathcal{F}_n[f(\mathbf{r})] \text{ and } N(\mathbf{k}) = \mathcal{F}_n[n(\mathbf{r})]$$

E. Fundamental Cryptographic Law

Irrespective of the cryptographic model that is used, there is a fundamental law of cryptography which is a unifying principle in regard to securing against any successful attack. The law is that any encryption process should be undertaken using: *One message, one key, one algorithm*. This law underpins the use of a One-Time Pad (OTP), which yields ciphertexts that are:

(i) computationally secure, i.e. secure given an upper bound on the computational capabilities of an attacker;
(ii) unconditionally secure, i.e. secure whatever the computational power of the attacker.

It is in the context of this law that we consider the encryption/decryption of information using a (digital) phase-only approach.

The cryptographic law stated above can be extended to cases when it is required to hide the ciphertext in other data types, namely, a covertext. In this case, the law is extended to: *One message, one key, one algorithm, one covertext* which underpins the field of Steganocryptography [8]. The issue of using one algorithm to generate a cipher is not conventional as most algorithms require a significant input to be constructed, their numerical characteristics being fundamental to the cryptographic strength of the encryption system in which they are used and open to public scrutiny (usually within the cryptographic community). Constructing personalised encryption algorithms has been considered in [9], for example, which uses evolutionary computing to generate an encryption algorithm that approximates a finite stream of natural noise. As with the exchange of keys, there is the issue of how different algorithms can be exchanged. In regard to this issue, a three-way pass protocol is considered in Section IX of this work which, although focused on the exchange of keys used for execution of the encryption functions presented in this paper, can equally well be used to exchange a personalised encryption algorithm (or any other plaintext). In principle, point (ii) above includes the use of quantum computers and underpins the field of post-quantum cryptography as brifly discussed on the following section.

F. Post-quantum Cryptography

The cryptographic strength of an encryption algorithm usually reflects on how difficult it is to break a cipher and has traditionally been evaluated in terms of measures that are related to the conventional computing power available at the time the algorithm was designed. It has been known for some time that the majority of popular public-key encryption algorithms can be efficiently broken by a sufficiently powerful quantum computer. However, until recently, quantum computing was a hypothetical concept and so the idea of breaking legacy encryption algorithms using quantum computers was only of theoretical interest. Basic quantum computers now exist in many research laboratories across the world and a range of companies and governments have research programs in the field. For this reason, post-quantum cryptography, which is concerned with the development of encryption algorithms that are considered secure even against an attack by a quantum computer, is starting to come of age [10]. Thus, new projects such as the 'Open Quantum Safe' project, initiated in late 2016 [11], have been established with the goal of developing quantum-resistant encryption algorithms. This includes the development of an open source library of post-quantum schemes and C functions for quantum-resistant encryption algorithms with an initial focus on key exchange algorithms. This is because legacy key exchange algorithms are particularly vulnerable to quantum computers, a consequence that relates to the legacy of using modular arithmetic with prime numbers and that all modular arithmetic functions are characterised be repeating periods.

Quantum computers are destined to make legacy encryption methods redundant although a realistic time scale for this is not clear. One of the principle concerns is the potential redundancy of the RSA algorithm which can be 'broken' by application of the Shor algorithm [12]. Originally developed by Peter Shor at Massachusetts Institute of Technology in 1994 for prime number factorisation, the algorithm leads to an exponentiation of the computing time required as the prime numbers (whose factors are required) increases in size due primarily to the period finding step in Shor's algorithm. However, used in conjunction with a quantum computer, and, by exploiting the properties of the quantum Fourier transform, the algorithm can be implemented very effectively, the problem becoming equivalent to the quantum phase estimation problem [13]. In addition to having the potential to make many legacy digital encryption methods redundant, quantum computing has the potential to provide new methods for encrypting data. This includes dealing with quantum multi-level secret sharing schemes to encrypt quantum states with access structures for decryption [14], for example. In this context, phase-only encryption is one of a number of approaches that is crucially not related to prime number based cryptography; it is a floating point based form of encryption and has application to key-exchange as discussed in Section IX. In this sense, the work reported in this paper may be considered to play a part in the arena of post-quantum cryptography, although a detailed exposition of this lies beyond the scope of the current work.

III. Deconvolution for Phase-only Functions

Given Equation (4), it is clear that in order to decrypt the signal $s(\mathbf{r})$ to recover $f(\mathbf{r})$, it is necessary to deconvolve $s(\mathbf{r})$ given the cipher $n(\mathbf{r})$. It is well known that the deconvolution problem is, in generally, an ill-posed problem with a wide range of primarily conditional solutions. Application of a phase-only cipher solves this issue thereby allowing both encryption and decryption to be performed efficiently subject to the encryption model compounded in Equation (4). This result is the fundamental 'key' to phase-only encryption, as compounded in the following theorem.

Theorem III.1. *If $n(\mathbf{r})$ has a phase-only spectrum, then the deconvolution problem associated with Equation (4) is well-posed, i.e. $f(\mathbf{r})$ can be recovered from $s(\mathbf{r})$ exactly and uniquely, except for the value of $f(\mathbf{r} = \mathbf{0})$ which remains undefined.*

Proof: Let $\Theta(\mathbf{k})$ be the phase function of a unit amplitude phase-only spectrum such that

$$\exp[i\Theta(\mathbf{k})] = N(\mathbf{k}) = \mathcal{F}_n[n(\mathbf{r})]$$

Applying the convolution theorem to Equation (4), we can write

$$S(\mathbf{k}) = \exp[i\Theta(\mathbf{k})]F(\mathbf{k}) + \exp[i\Theta(\mathbf{k})]$$

and it is then clear that

$$\exp[-i\Theta(\mathbf{k})]S(\mathbf{k}) = F(\mathbf{k}) + 1$$

Hence, using the correlation theorem (where $\odot$ denotes the n-dimensional correlation integral) and Equation (1) we have

$$f(\mathbf{r}) + \delta^n(\mathbf{r}) = n^*(\mathbf{r}) \odot s(\mathbf{r}) \equiv \int_{-\infty}^{\infty} n^*(\mathbf{r}+\mathbf{r}')s(\mathbf{r}')d^n\mathbf{r}'$$

However, since the function $\delta^n(\mathbf{r})$ only effects the value of $f(\mathbf{r})$ at $\mathbf{r} = \mathbf{0}$ and has no influence for all values of $r > 0$, we can write

$$f(\mathbf{r}) = n^*(\mathbf{r}) \odot s(\mathbf{r}),\ r > 0$$

or alternative, setting the value for $f(\mathbf{r})$ to zero at $r = 0$ (which we call the 're-normalising condition'),

$$f(\mathbf{r}) = n^*(\mathbf{r}) \odot s(\mathbf{r}),\ f(\mathbf{r} = \mathbf{0}) = 0 \tag{5}$$

■

Corollary III.1. *Given that*

$$S(\mathbf{k}) = \exp[i\Theta(\mathbf{k})]F(\mathbf{k}) + \exp[i\Theta(\mathbf{k})]$$

and since $\|\exp[i\Theta(\mathbf{k})]\|_\infty = 1$*, then using Minkowski's inequality [15] for the sum of two functions, it follows that*

$$\|S(\mathbf{k})\|_\infty \leq 1 + \|F(\mathbf{k})\|_\infty$$

Corollary III.2. *Let* $f(\mathbf{r})\ \exists\ \forall \mathbf{r} \in \mathbb{R}^n$ *be of compact support* $\mathbb{R}^n$*. If we then consider the ratio*

$$R = \frac{\|n(\mathbf{r}) \otimes f(\mathbf{r})\|_p}{\|n(\mathbf{r})\|_p}$$

it is clear that $R \geq 0$ *is a measure of the Signal-to-Noise Ratio (SNR) since, using Minkowski's inequality together with Young's inequality for the convolution of two functions [16],*

$$\|s(\mathbf{r})\|_p = \|n(\mathbf{r}) \otimes f(\mathbf{r}) + n(\mathbf{r})\|_p$$

$$\leq \|n(\mathbf{r}) \otimes f(\mathbf{r})\|_p + \|n(\mathbf{r})\|_p \leq \|n(\mathbf{r})\|_p \times \|f(\mathbf{r})\|_p + \|n(\mathbf{r})\|_p$$

leading to the inequalities

$$\frac{\|s(\mathbf{r})\|_p}{\|n(\mathbf{r})\|_p} \leq 1 + R \text{ and } R \leq \|f(\mathbf{r})\|_p$$

Hence, if we consider the equation

$$s(\mathbf{r}) = n(\mathbf{r}) \otimes f(\mathbf{r}) + cn(\mathbf{r})$$

where $c = (1 + R)^{-1}$ *then, from Theorem III.1, it is clear that Equation (5) can be generalised to the form*

$$f(\mathbf{r}) = n^*(\mathbf{r}) \odot s(\mathbf{r}),\ f(\mathbf{r} = \mathbf{0}) = 0,\ \forall\ c > 0$$

This means that $f(\mathbf{r}),\ \forall r > 0$ *can be uniquely recovered from* $s(\mathbf{r})$ *whatever the magnitude of the SNR which we define as* $1 + R$*, i.e. however small the value of the SNR becomes or equivalently, however large the value of* c *becomes.*

Remark III.1. Corollary III.2 is a theoretical result associated with piecewise continuous functions, and, the statement 'however large the value of c becomes' is *null and void* for applications involving numerical operations on finite discrete arrays. As discussed later on in this paper (in Section VIII.B), the value of c has an upper bound subject to the floating point precision available, i.e. the numerical value of c used places limits on the accuracy to which a discrete form of $f(\mathbf{r})$ can be deconvolved from a discrete version f $s(\mathbf{r})$, both functions being taken to be Nyquist sampled.

Remark III.2. The result compounded in Theorem III.1 and Corollary III.2 is of little value to the deconvolution problem in general which occurs in the applications of digital signal and image processing, for example. This is because it can not be assumed that the IRF has a stochastic phase-only spectrum, i.e. $P(\mathbf{k}) = \mathcal{F}_n[p(\mathbf{r})] \neq N(\mathbf{k}) = \exp[i\Theta(\mathbf{k})]$. Further, natural noise can not, in general, be assumed to be characterised by phase-only functions. Thus it should be understood that Theorem III.1 is strictly only applicable to the convolution model given in Equation (4) when $n(\mathbf{r})$ has a phase-only spectrum and has no applicability to signal processing in general. However, the result given in Corollary III.2 does have applications in the area of cryptography. This is because the diffused plaintext $n(\mathbf{r}) \otimes f(\mathbf{r})$ can be completely embedded in a (phase-only) cipher $cn(\mathbf{r})$ (for $c >> 1$) thereby fully dissipating the statistical signature of the diffused plaintext in the distribution of the cipher.

Remark III.3. Theorem III.1 can be generalised further by considering an encryption model of the form

$$s(\mathbf{r}) = m(\mathbf{r}) \otimes f(\mathbf{r}) + n(\mathbf{r})$$

where $m(\mathbf{r})$ is a secondary phase-only stochastic function such that

$$m(\mathbf{r}) \leftrightarrow \exp[i\Phi(\mathbf{k})],\ \ \Phi(\mathbf{k}) \neq \Theta(\mathbf{k})$$

In this case, $F(\mathbf{k})$ is given by (following the proof of Theorem III.1 as given)

$$F(\mathbf{k}) = \exp[-i\Phi(\mathbf{k})]S(\mathbf{k}) - \exp[-i\Phi(\mathbf{k})]\exp[i\Theta(\mathbf{k}))]$$

and, via the correlation theorem,

$$f(\mathbf{r}) = m^*(\mathbf{r}) \odot s(\mathbf{r}) - m^*(\mathbf{r}) \odot n(\mathbf{r})$$

Now, providing $m(\mathbf{r})$ is statistically independent of $n(\mathbf{r})$, then in the asymptotic limit $r \to \infty$, we can expect that $m^*(\mathbf{r}) \odot n(\mathbf{r}) \to 0$. However, this result is not as viable as is the case when $m(\mathbf{r}) = n(\mathbf{r})$ considered in Theorem III.1. This is because in Theorem III.1, the deconvolution for $f(\mathbf{r})$ is exact and unique $\forall r > 0$. In the case when $m(\mathbf{r}) \neq n(\mathbf{r})$, the deconvolution for $f(\mathbf{r})$ can, in practice, be expected to be characterised by additive noise associated with the correlation function $m^*(\mathbf{r}) \odot n(\mathbf{r})$ and is therefore an exact but not a unique result, i.e. $f(\mathbf{r})$ is exactly recovered via the correlation function $m^*(\mathbf{r}) \odot s(\mathbf{r})$ but is non-unique due to the perturbation of this function by the function $m^*(\mathbf{r}) \odot n(\mathbf{r})$.

IV. Encryption using Phase-Only Stochastic Functions

Consider a generic encryption model based on the linear stationary convolution equation for plaintext $f(\mathbf{r})$, cipher $n(\mathbf{r})$ and ciphertext $s(\mathbf{r})$ given by

$$s(\mathbf{r}) = n(\mathbf{r}) \otimes f(\mathbf{r}) + cn(\mathbf{r}) \tag{6}$$

where c is a real positive constant and

$$n(\mathbf{r}) \leftrightarrow\ \exp[i\Theta(\mathbf{k})],\ \Theta(\mathbf{k}) \in [-\pi, \pi]$$

noting that $n(\mathbf{r})$ is a complex function but that $f(\mathbf{r})$ may be real or complex. We consider the function $n(\mathbf{r})$ to be a cipher generated by some key dependent algorithm characterised by a phase-only spectrum with a random phase function $\Theta(\mathbf{k})$

and PDF $\Pr[n(\mathbf{r})]$. In this context, we refer to the convolution operation $n(\mathbf{r}) \otimes f(\mathbf{r})$ as the process of stochastic diffusion and the addition of noise through the term $cn(\mathbf{r})$ as the process of stochastic confusion, terms which have previously been introduced and discussed Section II.D. For a plaintext function $f(\mathbf{r})$ that is taken to be real, the ciphertext is taken to be given by $\mathrm{Re}[s(\mathbf{r})]$ and has some PDF $\Pr\{\mathrm{Re}[s(\mathbf{r})]\}$.

In Fourier space, Equation (6) becomes

$$S(\mathbf{k}) = F(\mathbf{k}) \exp[i\Theta(\mathbf{k})] + c \exp[i\Theta(\mathbf{k})] \tag{7}$$

and it is clear that the value of c controls the magnitude of the term $\exp[i\Theta(\mathbf{k})]$ compared with the term $F(\mathbf{k}) \exp[i\Theta(\mathbf{k})]$ and as c increases in magnitude, the spectrum $S(\mathbf{k})$ becomes dominated by the phase-only spectrum $\exp[i(\Theta(\mathbf{k})]$. Thus by choosing a large value for c (which is quantified later) we can 'embed' the spectrum $F(\mathbf{k}) \exp[i\Theta(\mathbf{k})]$ in the phase-only spectrum $\exp[i\Theta(\mathbf{k})]$ in the knowledge that $F(\mathbf{k})$ can be recovered exactly as a consequence of Corollary III.2. The constant c is a measure of SNR^{-1} which we refer to as the 'Spectral Embedding Coefficient'. It may be classified in terms of a conventional Decibel scale for the SNR via the equation

$$c_{dB} = 10 \log_{10}(c) \tag{8}$$

This result is invariant of the stochastic phase function $\Theta(\mathbf{k})$ that is chosen for $c >> 1$. The output will be dominated by the stochastic behaviour associated with the PDF of $n(\mathbf{r})$ thereby 'eliminating', in the output, the 'statistical signature' associated with the term $n(\mathbf{r}) \otimes f(\mathbf{r})$ which is dependent on the information function $f(\mathbf{r})$. Thus the distribution of the plaintext is well dissipated in the distribution of the cipher through both convolution (diffusion) and addition (confusion). However, for $c >> 1$, addition of the cipher is the dominating effect, maximising confusion which in practice, is subject to the floating point accuracy required to give a decrypt with an acceptable accuracy (to be quantified in Section VIII.B). It is this observation that is fundamental to using Theorem III.1 for encryption, yields the algorithms considered in Section VII and can be considered to be a form of data hiding, e.g. [19], [20], [21] and [22].

V. Statistical Analysis

Given that the function $s(\mathbf{r})$ in Equation (6) is the sum of two terms, it is informative to investigate the expected PDFs of the real parts of these terms, i.e. the real parts of $f(\mathbf{r}) \otimes n(\mathbf{r})$ and $n(\mathbf{r})$, noting that $f(\mathbf{r})$ is assumed to be real and $\mathrm{Re}[n(\mathbf{r})]$ is taken to have a PDF $\Pr\{\mathrm{Re}[n(\mathbf{r})]\}$ with at least a finite variance.

Consider the first term and the application of a δ^n-sequence model for $f(\mathbf{r})$ given by

$$\begin{aligned} f(r_1, ..., r_n) = & \sum_{m_1=-\infty}^{\infty} ... \sum_{m_n=-\infty}^{\infty} c(m_1, ..., m_n) \\ & \times \delta(r_1 - m_1)...\delta(r_n - m_n) \end{aligned} \tag{9}$$

where $f(\mathbf{r})$ is taken to be a piecewise continuous function and $c(m_1, ..., m_n)$ are real coefficients. Taking the PDF to be a function of the independent variable ξ, say, application of the Central Limit Theorem (CLT) yields

$$\begin{aligned} & \Pr\{f(\mathbf{r}) \otimes \mathrm{Re}[n(\mathbf{r})]\}(\xi) \equiv \\ & \Pr\{f(r_1, ..., r_n) \otimes \mathrm{Re}[n(r_1, ..., r_n)]\}(\xi) \\ & = \Pr\Bigg\{ \sum_{m_1=-\infty}^{\infty} ... \sum_{m_n=-\infty}^{\infty} c(m_1, ..., m_n) \\ & \times \mathrm{Re}[n(r_1 - m_1, ..., r_n - m_n)] \Bigg\}(\xi) \\ & = \Pr\{\mathrm{Re}[n(r_1, ..., r_n)]\} \otimes \Pr\{\mathrm{Re}[n(r_1, ..., r_n)]\} \otimes ... \\ & \sim \mathrm{Gauss}(\xi) \end{aligned} \tag{10}$$

where $\mathrm{Gauss}(\xi)$ denotes a generic Gaussian distribution which is obtained in the asymptotic limit as the number of convolution of the same PDF tends to infinity [23]. Hence, we can expect the PDF of the $\mathrm{Re}[n(\mathbf{r}) \otimes f(\mathbf{r})]$ to be normally distributed irrespective of the PDF of the stochastic phase function $\Theta(\mathbf{k})$. Note that the above result comes from the principal theme of the CLT, namely, that when independent random variables are added, their sum (assuming proper normalisation) tends toward a normal distribution even if the original variables themselves are not normally distributed so that for independent real stochastic functions $n_1(\mathbf{r}), n_2(\mathbf{r}), ...$, say,

$$\begin{aligned} \Pr[n_1(\mathbf{r}) + n_2(\mathbf{r}) + ...](\xi) = \Pr[n_1(\mathbf{r})](\xi) \otimes \Pr[n_2(\mathbf{r})](\xi) \otimes \\ ... \sim \mathrm{Gauss}(\xi) \end{aligned}$$

Although Equation (10) is predicated on a single real stochastic function $\mathrm{Re}[n(\mathbf{r})]$ with a specific PDF, the δ^n-sequence model for $f(\mathbf{r})$ given by Equation (9) yields an infinite set of shifted stochastic functions. In this context $n_1(\mathbf{r}), n_2(\mathbf{r}), ...$ are shifted versions of $\mathrm{Re}[n(\mathbf{r})]$.

In regard to the second term of Equation (6) given by $cn(\mathbf{r}) \leftrightarrow c[\exp[i\Theta(\mathbf{k})]]$, $\mathrm{Re}[n(\mathbf{r})]$ is a composite sum of independent random variables whose empirical distribution is therefore, through application of the CLT, a normal distribution [24]. Thus we may expect that

$$\Pr\{\mathrm{Re}[n(\mathbf{r})]\}(\xi) \sim \mathrm{Gauss}(\xi)$$

We can also expect that

$$\Pr\{\mathrm{Im}[n(\mathbf{r})]\}(\xi) \sim \mathrm{Gauss}(\xi)$$

and if we relax the condition on taking the real component of $n(\mathbf{r})$, then we obtain the complex PDF $\Pr\{\mathrm{Re}[n(\mathbf{r})], \mathrm{Im}[n(\mathbf{r})]\}$. Hence, given that both the real and imaginary components are normally distributed, the joint distribution in the complex plane will be given by a Rayleigh distribution and we can therefore expect that

$$\Pr\{\mid n(\mathbf{r}) \mid\}(\xi) \sim \mathrm{Rayleigh}(\xi)$$

In summary, given Equation (6), we expect the PDFs of the real (and imaginary) components of both terms on the RHS of this equation to be normally distributed in the asymptotic limit. Thus the signal $\mathrm{Re}[s(\mathbf{r})]$ can be expected to be normally distributed and the PDF of the signal $\mid s(\mathbf{r}) \mid$ to be Rayleigh distributed.

VI. Cryptanalysis

We consider some approaches to breaking the cipher $s(\mathbf{r})$ in Equation (6). Two cases are considered, namely, a Bayesian and a correlation attack.

A. Bayesian based Cryptanalysis

It is well known that Bayesian analysis can be used to generate a Maximum Likelihood (ML) estimate for $f(\mathbf{r})$ given Equation (2) where a Gaussian PDF for $n(\mathbf{r})$ can be applied, a statistical model that is relevent given the analysis provided in Section V. In this case, the estimate for $f(\mathbf{r})$, as shown in Appendix A, is given by

$$f(\mathbf{r}) = g^*(\mathbf{r}) \odot s(\mathbf{r}) \text{ where } g^*(\mathbf{r}) \leftrightarrow G^*(\mathbf{k}) = \frac{N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}$$

and with $N(\mathbf{k}) = \exp[i\Theta(\mathbf{k})]$ it is clear that

$$G^*(\mathbf{k}) = \exp[-i\Theta(\mathbf{k})]$$

This result rules out the ability to develop a Bayesian attack because:

(i) the statistical signature of the ciphertext is dominated by the (Gaussian distributed) function $\mathrm{Re}[n(\mathbf{r})]$ (providing $c >> 1$) and the statistical signature of the plaintext $f(\mathbf{r})$, after diffusion with $n(\mathbf{r})$, is therefore not available.

(ii) Even though the distribution of $\mathrm{Re}[n(\mathbf{r})]$ is Gaussian, the stochastic phase function $\Theta(\mathbf{k})$ can be conditioned to be uniformly distributed as discussed in Section VII - see Equation (15). Thus, the homomorphic equation

$$\ln F(\mathbf{k}) = -i\Theta(\mathbf{k}) + \ln S(\mathbf{k})$$

obtained by taking logarithms of the ML estimate in Fourier space does not provide a solution that is statistically significant when

$$\frac{\partial}{\partial \xi}\Pr[\Theta(\mathbf{k})](\xi) = 0$$

B. Correlation based Cryptanalysis

Consider Equation (6). Given that $s(\mathbf{r})$ is known, we can construct the correlation function $s^*(\mathbf{r}) \odot s(\mathbf{r})$. In Fourier space, this yields an equation for the power spectrum of the ciphertext given by

$$\begin{aligned} \mid S(\mathbf{k}) \mid^2 &= \mid F(\mathbf{k}) \exp[i\Theta(\mathbf{k})] + c \exp[i\Theta(\mathbf{k})] \mid^2 \\ &= \mid F(\mathbf{k}) \mid^2 + cF(\mathbf{k}) + cF^*(\mathbf{k}) + c^2 \\ &= \mid F(\mathbf{k}) \mid^2 + 2c\mathrm{Re}[F(\mathbf{k})] + c^2 \\ &= \mid F(\mathbf{k}) \mid^2 + c^2 \left[1 + \frac{2}{c}\mathrm{Re}[F(\mathbf{k})]\right] \end{aligned}$$

and hence, it is clear that

$$\mid S(\mathbf{k}) \mid^2 = \mid F(\mathbf{k}) \mid^2 + c^2,\ c \to \infty$$

or, using the correlation theorem,

$$s^*(\mathbf{r}) \odot s(\mathbf{r}) = f^*(\mathbf{r}) \odot f(\mathbf{r}) + c^2\delta^n(\mathbf{r});\ c \to \infty$$

and thus

$$s^*(\mathbf{r}) \odot s(\mathbf{r}) = f^*(\mathbf{r}) \odot f(\mathbf{r});\ r > 0,\ c \to \infty$$

This result illustrates why it is important to apply a large value of c (within the floating point accuracy available in practice) to encrypt the information function $f(\mathbf{r})$, thereby eliminating the potential for an iterative attack based on solving the quadratic equation

$$\mathrm{Re}[F(\mathbf{k})]^2 + \mathrm{Im}[F(\mathbf{k})]^2 + 2c\mathrm{Re}[F(\mathbf{k})] + [c^2 - \mid S(\mathbf{k}) \mid^2] = 0$$

for Re[F(**k**)] when $\mathrm{Im}[F(\mathbf{k})]$ and c are undefined. Nevertheless, when $c \to \infty$, autocorrelating the ciphertext leads to a decryption problem that is equivalent to the phase retrieval problem, i.e. given that $f(\mathbf{r}) \leftrightarrow F(\mathbf{k}) = \mid F(\mathbf{k}) \mid \exp[i\Phi(\mathbf{k})]$, where $\Phi(\mathbf{k})$ is the phase function of the plaintext (not the cipher), then if and only if $\mid F(\mathbf{k}) \mid$ is known, we are required to estimate $\Phi(\mathbf{k})$, upon which $f(\mathbf{r})$ can be obtained by Fourier inversion.

In general, the phase retrieval problem is severely ill-posed with no uniformly stable solutions in infinite-dimensional spaces, a result that holds for frames that are continuous. However, for $\mathbf{r} \in \mathbb{R}^2$, phase estimation algorithms have been developed to provide approximate solutions to this problem, especially in regard to X-ray crystallography, for example, where the magnitude only diffraction pattern in the far-field is determined by the two-dimensional Fourier transform of the diffractor, i.e. the crystal, [2]. In this context, correlation of the ciphertext could lead to a phase retrieval based attack using algorithms described in [28] and [29], for example. However, there is a simple solution to preventing such an attack which is to replace the Spectral Embedding Coefficient c with $c\Theta(\mathbf{k})$ thereby weighting the phase-only spectrum with the phase function. This is because in Fourier space, Equation (6) can be modified to

$$S(\mathbf{k}) = F(\mathbf{k}) \exp[i\Theta(\mathbf{k})] + c\Theta(\mathbf{k}) \exp[i(\Theta(\mathbf{k})] \tag{11}$$

when $F(\mathbf{k})$ is now given by

$$F(\mathbf{k}) = \exp[-i\Theta(\mathbf{k})]S(\mathbf{k}) - c\Theta(\mathbf{k}) \tag{12}$$

and, via the correlation theorem

$$f(\mathbf{r}) = n^*(\mathbf{r}) \odot s(\mathbf{r}) - c\theta(\mathbf{r})$$

where $\theta(\mathbf{r}) = \mathcal{F}_n^{-1}[\Theta(\mathbf{k})]$. The power spectrum of the ciphertext is then given by (noting the $\Theta(\mathbf{k})$ is a real function)

$$\mid S(\mathbf{k}) \mid^2 = \mid F(\mathbf{k}) \mid^2 + c^2\Theta^2(\mathbf{k}),\ c \to \infty \tag{13}$$

with application of the correlation and convolution theorems yielding the equation

$$s^*(\mathbf{r}) \odot s(\mathbf{r}) = f^*(\mathbf{r}) \odot f(\mathbf{r}) + c^2\theta(\mathbf{r}) \otimes \theta(\mathbf{r})$$

It is then clear that the phase retrieval problem can not be attempted with out access to knowledge of the cipher $\Theta(\mathbf{k})$, i.e. the same cipher that is used to phase-only encrypt the plaintext. We call this solution 'Weighted Phase-only Encryption' (WPOE). In principle, a secondary pseudo random number generating algorithm can be used so that Equations (11) and (12) are replaced with

$$S(\mathbf{k}) = F(\mathbf{k}) \exp[i\Theta_1(\mathbf{k})] + c\Theta_2(\mathbf{k}) \exp[i(\Theta_1(\mathbf{k})]$$

and

$$F(\mathbf{k}) = \exp[-i\Theta_1(\mathbf{k})]S(\mathbf{k}) - c\Theta_2(\mathbf{k}) \tag{14}$$

respectively, where $\Theta_1(\mathbf{k})$ and $\Theta_2(\mathbf{k})$ are independent ciphers generated by the same algorithm with different keys or two different algorithms with the same or different keys. A secondary algorithm/key is then required to decrypt the ciphertext using Equation (14) in order to secure against an attack through analysis of ciphertext correlation and the implementation of a phase retrieval algorithm.

In the following section, algorithms are developed that provide a user with two options:

Option (1): Phase-only Encryption (POE) - encryption based on Equation (7) where decryption is not dependent on knowledge of c only knowledge of the key(s) used to compute $\Theta(\mathbf{k})$.

Option (2): Weighted Phase-only Encryption (WPOE) - encryption based on Equation (11) where decryption is dependent on knowledge of c as well as knowledge of the key(s).

VII. Phase-only Encryption/Decryption Algorithms

Although the terms $\mathrm{Re}[p(\mathbf{r}) \otimes f(\mathbf{r})]$ and $\mathrm{Re}[n(\mathbf{r})]$ from Equation (6) can be expected to have non-uniform distributions (as discussed in Section V and as illustrated in Figures 1 and 2 of Section VIII), the stochastic phase function $\Theta(\mathbf{k})$ is assumed to be, or, at least can be conditioned to be uniformly distributed and wrapped between $-\pi$ and π radians, i.e.

$$\Pr[\Theta(\mathbf{k})](\xi) = U(\xi),\ \xi \in [-\pi, \pi] \tag{15}$$

where

$$U(\xi) = \begin{cases} \frac{1}{2\pi}, & \forall \xi \in [-\pi, \pi]; \\ 0, & \forall \xi \notin [-\pi, \pi]. \end{cases}$$

There are two principal ways to compute the random phase function $\Theta(\mathbf{k}) \in [-\pi, \pi]\ \forall \mathbf{k}$:

(i) Generate a normalised uniformly distributed function $R(\mathbf{k}) \in [0, 1]$, say, and let

$$\Theta(\mathbf{k}) = 2\pi \left[R(\mathbf{k}) - \frac{1}{2} \right] \tag{16}$$

(ii) Construct $\Theta(\mathbf{k})$ via Fourier transformation of a random variable denoted by $r(\mathbf{r}) \in [0, 1]$ say, i.e.

$$\Theta(\mathbf{k}) = \mathrm{atan2}[R(\mathbf{k})],\ \ R(\mathbf{k}) = \mathcal{F}_n[r(\mathbf{r})] \tag{17}$$

where $\mathrm{atan2}$ yields the 4-quadrant phase values in the range $[-\pi, \pi]$ by computing one unique arc tangent value in which the signs of both arguments are used to determine the quadrant of the result, thereby selecting the desired branch of the arc tangent. Since $r(\mathbf{r})$ is real and has no symmetry, $R(\mathbf{k})$ has a symmetric real component and an asymmetric imaginary component. The 4-quadrant phase function is therefore an asymmetric function, i.e. $\Theta(\mathbf{k}) = -\Theta(-\mathbf{k})$, and has a symmetric uniform PDF. In this context, the stochastic signature of $\Theta(\mathbf{k})$ in one half-space is repeated in the other. The two half-spaces are therefore correlated.

One of the principal differences between the two approaches for computing the phase function given above is that application of Equation (16) requires the complex ciphertext to be used in order to recover the plaintext, whereas Equation (17) requires only the real component of the ciphertext to be retained thereby halving the data storage/transmission requirements needed to recover the plaintext albeit at the 'price' of using a stochastic phase function with the property $\Theta(\mathbf{k}) = -\Theta(-\mathbf{k})$. For this reason, in the following sections, Equation (17) is used, the principal steps associated with the two algorithms being presented for $\mathbf{r} \in \mathbb{R}^1$ without considering any data or data processing error checks. In the following step-by-step algorithms, the two options quantified at the end of Section VI are given in 'Step 4', the notation used being chosen to reflect that of the preceding mathematical analysis for the case of a one-dimensional discrete array.

A. Phase-only Encryption Algorithm

Inputs: Plaintext array f_n (floating point reals) of size N (integer) read from a plaintext file; real constants $key > 0$ (integer), $c > 0$ (floating point); opt (integer) - $opt = 0$ provides POE, $opt \neq 0$ provides WPOE.

Data processing functions: $Algol$ - algorithm for generating an array of (real) random numbers for a known key key; Discrete Fourier Transform (DFT) via application of a Fast Fourier Transform (FFT) algorithm.

Step 1: For an input data stream $f_n,\ \ n = 1, 2, .., N$, compute a uniformly distributed random variable array $r_n \in [0, 1],\ \ n = 1, 2, ..., N$ using a known algorithm $Algol(key)$ seeded by a known key key.

Step 2: Compute the discrete (complex) spectrum R_n of r_n using a DFT.

Step 3: Compute the phase-only spectrum $\exp(i\Theta_n)$ by returning the 4-quadrant phase angles $\Theta_n \in [-\pi, \pi]$ of R_n.

Step 4:

(i) For $opt = 0$, compute the array

$$S_n = F_n \exp(i\Theta_n) + c \exp(i\Theta_n)$$

(ii) For $opt \neq 0$, compute the array

$$S_n = F_n \exp(i\Theta_n) + c\Theta_n \exp(i\Theta_n)$$

Step 5: Compute the inverse DFT of S_n, returning $\mathrm{Re}[s_n]$.

Output: Ciphertext array s_n (floating point reals) of size N (integer) written to a ciphertext file.

Remark VII.1. Note that the value of c is limited by the floating point accuracy used. Within the floating point precision applied, if the value of c becomes excessive, then floating point truncation of the array $F_n \exp(i\Theta_n)$ relative to $c \exp(i\Theta_n)$ or $c\Theta_n \exp(i\Theta_n)$ will occur, and thus, decryption of the array s_n (as given in the following algorithm) becomes subject to floating point error. For $c = 10^n$, a floating point precision of at least n decimal places is required.

B. Phase-only Decryption Algorithm

Data inputs: Ciphertext array s_n (floating point reals) of size N (integer) read from a ciphertext file, real constant $key > 0$ (integer), $c > 0$ (floating point), opt (integer) - $opt = 0$ provides Phase-only Decryption (POD) when

knowledge of c used for POE is not required, $opt \neq 0$ provides Weighted Phase-only Decryption (WPOD) when knowledge of c used for WPOE is required.

Data processing functions: $Algol$ - outputs an array of (real) random numbers for an identical key key to that used for encryption; Discrete Fourier Transform (DFT) via application of a Fast Fourier Transform (FFT).

Step 1: For the input ciphertext array $s_n,\ n = 1, 2, .., N$ (taken to be composed on N reals), compute random variable array $r_n \in [0.1],\ n = 1, 2, ..., N$ using an known algorithm $Algol(key)$ seeded by key key - same algorithm/key used to generate the ciphertext s_n.

Step 2: Compute the discrete (complex) spectrum R_n of r_n using the DFT.

Step 3: Compute the phase-only spectrum $\exp(i\Theta_n)$ by returning the phase angles $\Theta_n \in [-\pi, \pi]$ of R_n.

Step 4:

(i) For $opt = 0$, compute the array

$$F_n = S_n \exp(-i\Theta_n)$$

(ii) For $opt \neq 0$, compute the array

$$F_n = S_n \exp(-i\Theta_n) - c\Theta_n \exp(-i\Theta_n)$$

Step 5: Compute the inverse DFT of F_n and return the real components f_n setting $f_1 = 0$ for $opt = 0$ - re-normalising condition as specified in Equation (5).

Output: Plaintext array f_n (floating point reals) of size N (integer) written to a plaintext file.

Remark VII.2. It is assumed that the key(s) are subject to the usual conditions associated with high strength cryptography in terms of key length and have binary representations that exhibit bit streams with a Binary Entropy Function $\simeq 1$.

Remark VII.3. $Algol(key)$ is assumed to be a cryptographically strong random number generating algorithm in terms of generating randomness with a uniform distribution, a uniform power spectral density function, a high Lyapunov exponent and high cycle length, for example. In this context, $Algol(key)$ should be personalised using an algorithm obtained through the application of Evolutionary Computing, e.g. [9], [25] and [17].

Remark VII.4. Although the algorithms given above are specific for a vector array, the principle is identical for processing multi-dimensional arrays. Prototype software designed for the case when $\mathbf{r} \in \mathbb{R}^2$ is considered in the following section.

VIII. MATLAB Prototype Code for $\mathbf{r} \in \mathbb{R}^2$

Appendix B provides prototype software using MATLAB to implement the algorithms given in Sections VII.A and VII.B but for two-dimensional array processing using the phase-only encryption function *POE(key_R,key_G, key_B,c,opt)* and the phase-only decryption function *POD(key_R,key_G, key_B,c,opt)* for a full colour RGB image (assumed to be a bitmap file - 'Plaintext.bmp'). Each RGB (8-bit) component of the input image is encrypted separately using an individual colour component key (i.e. *key_R, key_G* and *key_B*). The output for each colour component is written to a separate .txt file as a floating point matrix using the MATLAB function *dlmwrite* with a specified precision. The value of c is specified as an input to both the encryption and decryption functions. However, for $opt = 0$, the value of c used to decrypt is not required and c can be set to a dummy value (typically set $c = 1$) for the purpose of executing function *POD* in this case.

Both the encryption and decryption algorithms given in Appendix B rely on the use of the same uniformly distributed random number generator (MATLAB function *rand*) which is based on the Mersenne Twister algorithm, returning a uniformly distributed floating point number stream in the range 0-1 inclusively. However, this generator is used for illustrative purpose only, and, in practice, would be replaced with a cryptographically stronger algorithm as discussed in Remark VII.3. Each RGB component array is re-normalised separately after decryption according to Equation (5).

In terms of a conventional cryptographic protocol, the functions provided in Appendix B represent a floating point based symmetric encryption system in which Alice encrypts an image using three keys and transmits three files to Bob - typically in a zipped folder. Upon reception of these files, Bob uses the same keys to decrypt the data held in the three files. As with all other symmetric encryption schemes, this requires a key exchange protocol to be implemented such as the Diffie-Hellman [26] key exchange algorithm, or, as discussed in Section IX, the use of a phase-only approach for implementing a three-pass protocol. It is noted that the three keys plus three files 'solution' considered can be reduced to a single key plus single file protocol by concatenation. However, the three keys, three files solution provided can be used to yield a further level of security using 'time stamping'.

A. Test Case

Figure 1 shows an example of an input colour test image (input to function *POE*) and the (colour) decrypt (output of function *POD*) using RGB component keys, each composed of randomly generated integers consisting of a maximum of 10 digits (the limiting upper bound for a MATLAB random number generator with a non-negative integer seed $< 2^{32}$) for $c = 10^{10}$ and $opt = 0$ so that the term $c\exp[i\Theta(\mathbf{k})]$ in Equation (7) is ten orders of magnitude larger than the first term of the same equation. The RGB concatenated ciphertext is also displayed together with the normalised 256-bin histogram of the ciphertext obtained by taking the mean of the histograms for each re-normalised colour component (the 'RGB averaged' histogram) and setting the first component of the histogram to zero in order to eradicate the dominance of the 'black bin'. Figure 2 shows a similar example but for a different colour image encrypted for the case when $c = 10^{10}$ and $opt = 1$ thereby implementing WPOE compounded in Equations (11) and (12).

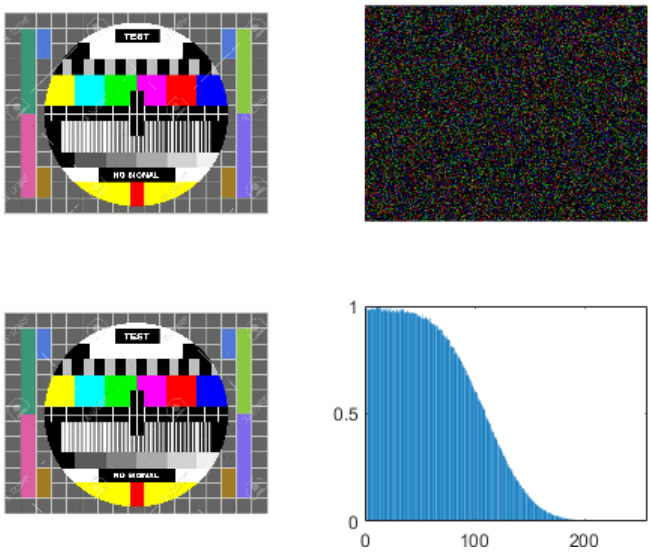

Fig. 1. Example application of the MATLAB functions given in Appendix B: plaintext (top-left), ciphertext (top-right), decrypt (bottom-left) and the normalised 256-bin RGB averaged histogram of the ciphertext (bottom-right) for $c = 0$ and $opt = 0$.

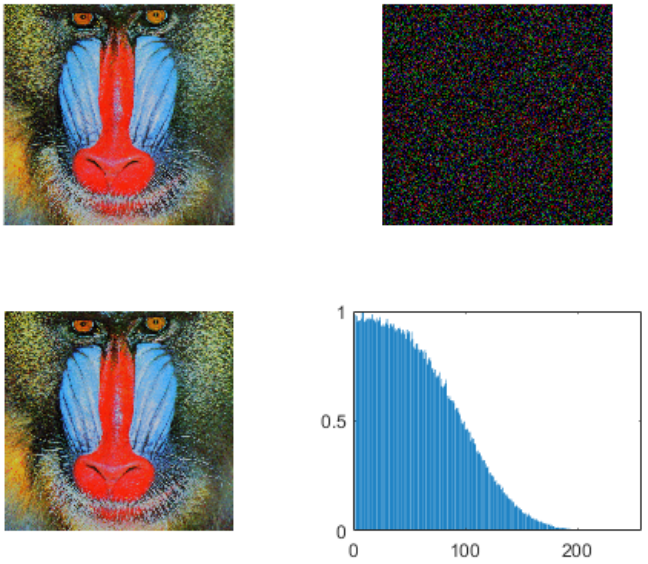

Fig. 2. Example application using the MATLAB functions given in Appendix B: plaintext (top-left), ciphertext (top-right), decrypt (bottom-left) and the normalised 256-bin RGB averaged histogram of the ciphertext (bottom-right) for $c = 10^{10}$ and $opt = 1$.

Figures 1 and 2 illustrates the difference in the distributions of the ciphertext between POE and WPOE. In the former case, the normalised histogram has a relatively uniform distribution over the central region and has relatively short tails compared to a standard normal distribution which is the statistical signature of the ciphertext for WPOE. This is due to the use of a uniformly distributed random number generator used to construct the phase function via Equation (17).

B. Numerical Analysis

The decrypt has a high fidelity with a Root Mean Square Error (RMSE) averaged over the RGB components of the order of 10^{-4}, the RMSE being defined as

$$\text{RMSE} = \frac{1}{3}(E_R + E_G + E_B) \tag{18}$$

where, for each colour component R, G and B,

$$E = \frac{1}{NM}\|P_{nm} - D_{nm}\|_2$$

Here, P_{nm} denotes the plaintext (input) image, D_{nm} denotes the decrypt (output) image (each array being composed of $N \times M$ pixels) and $\| \bullet \|_2$ denotes the Frobenius norm, i.e. for a two-dimensional array a_{nm}

$$\|a_{nm}\|_2 = \left(\sum_{n=1}^{N} \sum_{m=1}^{M} | \, a_{nm} \, |^2 \right)^{\frac{1}{2}}$$

However, as briefly discussed in Remark III.1, the accuracy of the decrypt (i.e. the value of the RMSE for a specific value of c) depends on the floating point precision that is applied, the operation of the algorithms provided in Appendix B being set to a default precision value of 32-bits, the MATLAB (2018a) default being 64 bits. This effect is quantified in Figure 3 which shows the log-log plots of the RMSE against values of the Spectral Embedding Coefficient $c \in [1, 10^{15}]$ for precisions of 4, 8 and 16 bits associated with the execution of function *POE* with $opt = 0$ (for the plaintext image given in Figure 1). This is achieved by changing the precision in the *dlmwrite* functions used at the end of function *POE* in Appendix B. The results given in Figure 3 illustrate the effect of decreasing the floating point precision. As the precision decreases, so does the maximum value of c_{dB} given by Equation (8) that can be applied. In each case, the RMSE remains constant until a threshold is reached, after which the RMSE increases linearly, the threshold decreasing with the precision applied.

Although the results given in Figure 3 have been obtained for the plaintext image given in Figure 1, they are indicative of all full colour image inputs, irrespective of size and format. The results show the numerical limits for which Theorem III.1 is operational in relation to the floating point accuracy under which the algorithms given in Appendix B are executed. It is therefore to be expected that as the floating point precision decreases so will the maximum value of c_{dB} that can be applied. By way of an example, for a precision of 8 bits, from Figure 3 (dotted line), it is clear that the largest value of $log_{10}c$ that can be applied before the RMSE starts to increase is the order of 6. Thus, from Equation (8), $\max[c_{\text{dB}}] \sim 60$ dB or equivalently, $\min[(1 + R)_{\text{dB}}] \sim -60$ dB (where $(1+R)_{\text{dB}} = 10\log_{10}(1+R) = -10\log_{10} c$ - see Corollary III.2) which is an exceptionally low minimum Signal-to-Noise Ratio subject to generating a decrypt which such high fidelity. Similarly, from Figure 3, for 16-bit precision the minimum SNR that can be applied is the order of -130 dB. These results are similar for the case when WPOE is applied (i.e. for $opt \neq 0$). However, this is specific to the plaintext image used in Figure 1 and for images of a different type (including format and size, for example) it should be expected that the results given in Figure 3 will change in scale but exhibit the same basic characteristics.

IX. Applications to Key Exchange using a Three-pass Protocol

The Three-Pass Protocol (TPP) is well known and a range of algorithms have been developed for its implementation. These include the Shamir TPP [30] and the Massey-Omura

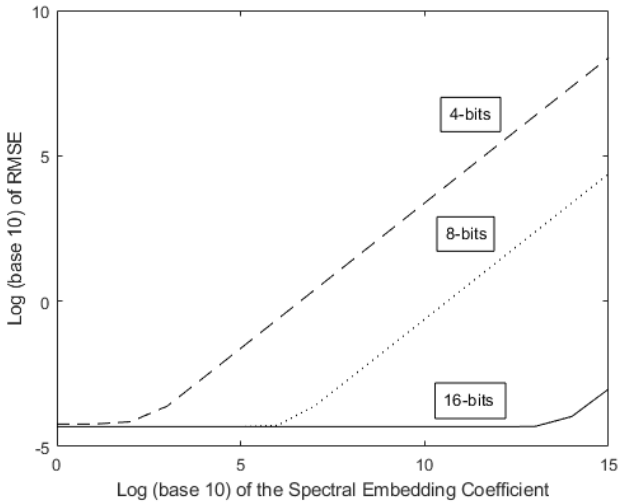

Fig. 3. Log-log (base 10) plots of the RMSE for values of the Spectral Embedding Coefficient c associated with the application of the functions given in Appendix B with bit precisions of 16 (solid line), 8 (dotted line) and 4 (dashed line) bits.

method [31]. The principle associated with the protocol is as follows: Alice encrypts her plaintext with a known algorithm and private key K_A say, and sends the ciphertext to Bob. Upon receipt of the ciphertext, Bob cannot decrypt the ciphertext because he does not know K_A. Instead Bob encrypts the ciphertext using the same algorithm and a new private key K_B and sends the now double encrypted plaintext back to Alice. Upon receipt, and critically, assuming the encryption algorithm is commutative, Alice can decrypt the double encrypted ciphertext with K_A and send the result (a single encrypted ciphertext) back to Bob who is then able to decrypt the result using K_B. By using this protocol, Alice and Bob do not need to agree upon K_A and K_B *a priori* and thus, no separate key exchange method is required. Three principal conditions for the application of this protocol are required:

- Irrespective of the number of encryptions that take place, the encryption algorithm used must be both commutative and strong enough so that it can not be broken using a known algorithm attack;
- the keys used must be of a sufficient length to make an exhaustive attack impracticable on any pass;
- if the encrypted information is intercepted for each of the three passes, it must not be possible to determine the plaintext from the three intercepts (assumed to be complete intercepts in each case).

In practice, it is the third of the conditions above that yields the greatest vulnerability and any encryption system that exploits this protocol must be based on algorithms that exhibit a 'computational difficulty'. For example, in the case of the Shamir and Massey-Omura algorithms, the security relies on the difficulty of computing discrete logarithms in a finite field [32]. In this section we consider an application of the three-pass protocol using a phase-only encryption (commutative) algorithm.

A. Basic Algorithm

Consider the classic TPP in which Alice wishes to exchange a single key given by the function $f(\mathbf{r}) \leftrightarrow F(\mathbf{k})$ where $f(\mathbf{r})$ is taken to be a real function, the value of the key for $f(\mathbf{r} = \mathbf{0})$ being taken to be redundant (re-normalisation condition). Alice generates the random phase cipher $\Theta_1(\mathbf{k})$ and similarly, Bob generates $\Theta_2(\mathbf{k})$ where both $\Theta_1(\mathbf{k})$ and $\Theta_2(\mathbf{k})$ are generated via application of the Fourier transform, i.e. $\exp[i\Theta_j(\mathbf{k})] \leftrightarrow n_j(\mathbf{r}),\ j = 1, 2$, the algorithm(s) for generating the ciphers $n_j(\mathbf{r})$ being taken to be cryptographically strong and ideally personal to Alice and Bob through application of an evolutionary computing approach [9]. The following steps are then applied.

Step 1: For a given value of $c >> 1$, known only to Alice, she encrypts $F(\mathbf{k})$ to produce ciphertext $S_1(\mathbf{k})$ using the equation

$$S_1(\mathbf{k}) = F(\mathbf{k}) \exp[i\Theta_1(\mathbf{k})] + c \exp[i\Theta_1(\mathbf{k})] \tag{19}$$

and sends $\mathrm{Re}[s_1(\mathbf{r})]$ of $s_1(\mathbf{r}) \leftrightarrow S_1(\mathbf{k})$ to Bob.

Step 2: Upon receiving the ciphertext $S_1(\mathbf{k}) \leftrightarrow \mathrm{Re}[s_1(\mathbf{r})]$, Bob encrypts $S_1(\mathbf{k})$ using the equation

$$S_2(\mathbf{k}) = S_1(\mathbf{k}) \exp[i\Theta_2(\mathbf{k})] \tag{20}$$

and sends $\mathrm{Re}[s_2(\mathbf{r})]$ of $s_2(\mathbf{r}) \leftrightarrow S_2(\mathbf{k})$ back to Alice.

Step 3: Alice decrypts Bobs ciphertext $S_2(\mathbf{k}) \leftrightarrow \mathrm{Re}[s_1(\mathbf{r})]$ using the equation

$$\begin{aligned} S_3(\mathbf{k}) &= S_2(\mathbf{k}) \exp[-i\Theta_1(\mathbf{k})] \\ &= S_1(\mathbf{k}) \exp[i\Theta_2(\mathbf{k})] \exp[-i\Theta_1(\mathbf{k})] \\ &= [F(\mathbf{k}) + c] \exp[i\Theta_1(\mathbf{k})] \exp[-i\Theta_1(\mathbf{k})] \exp[i\Theta_2(\mathbf{k})] \\ &= [F(\mathbf{k}) + c] \exp[i\Theta_2(\mathbf{k})] \end{aligned} \tag{21}$$

and sends $\mathrm{Re}[s_3(\mathbf{r})]$ of $s_3(\mathbf{r}) \leftrightarrow S_3(\mathbf{k})$ back to Bob.

Step 4: Bob decrypts the ciphertext $S_3(\mathbf{k}) \leftrightarrow \mathrm{Re}[s_3(\mathbf{r})]$ using the equation

$$F(\mathbf{k}) + c = S_3(\mathbf{k}) \exp[-i\Theta_2(\mathbf{k})] \tag{22}$$

The key is then given by $\mathrm{Re}[f(\mathbf{r})]\ r > 0$ where $f(\mathbf{r}) \leftrightarrow F(\mathbf{k})$ given that $\mathrm{Re}[f(\mathbf{0})]$ is undefined, or, with application of the re-normalisation condition $\mathrm{Re}[f(\mathbf{0})] = 0$.

B. Three-intercept Cryptanalysis

Assume that an attack is launched to estimate $f(\mathbf{r})$ based on knowledge of the TPP used (i.e. Steps 1-4 as given in Section IX.A) and knowledge of the functions $S_1(\mathbf{k})$, $S_2(\mathbf{k})$ and $S_3(\mathbf{k})$ obtained by intercepting the transmission associated with Steps 1-3 (and taking the Fourier transform of the results). Given Equations (19) - (22), we can eliminate the ciphers Θ_1 and Θ_2 to obtain the equation

$$F(\mathbf{k}) + c = \frac{S_1(\mathbf{k}) S_3(\mathbf{k})}{S_2(\mathbf{k})} = \frac{S_1(\mathbf{k}) S_2^*(\mathbf{k}) S_3(\mathbf{k})}{\mid S_2(\mathbf{k}) \mid^2}$$

or, from Equations (20),

$$F(\mathbf{k}) + c = \frac{S_1(\mathbf{k}) S_2^*(\mathbf{k}) S_3(\mathbf{k})}{\mid S_1(\mathbf{k}) \mid^2}$$

Further, given that the Spectral Embedding Coefficient c is taken to be set to high orders of magnitude (within the floating point precision available as discussed in Section VIII.B) then, from Equation (19), for the asymptotic case when $c \to \infty$

$$F(\mathbf{k}) + c = \frac{S_1(\mathbf{k})S_2^*(\mathbf{k})S_3(\mathbf{k})}{\mid F(\mathbf{k}) \mid^2 + c^2}$$

and it is clear that to obtain $F(\mathbf{k})$ we are required to solve the cubic equation

$$\begin{aligned} \mid F(\mathbf{k}) \mid^2 F(\mathbf{k}) + c \mid F(\mathbf{k}) \mid^2 + c^2 F(\mathbf{k}) + c^3 \\ = S_1(\mathbf{k})S_2^*(\mathbf{k})S_3(\mathbf{k}) \end{aligned} \tag{23}$$

for an unknown value of $c \to \infty$. Thus, the 'improbability' of 'breaking the key' lies in solving Equation (23) which can be written in terms of its constituent real and imaginary component via the following equations:

$$\begin{aligned} F_r^3(\mathbf{k}) + F^2(\mathbf{k})F_i(\mathbf{k}) + c[F_r^2(\mathbf{k}) + F_i^2(\mathbf{k})] + c^2 F_r(\mathbf{k}) + c^3 \\ = \mathrm{Re}[S_1(\mathbf{k})S_2^*(\mathbf{k})S_3(\mathbf{k})] \end{aligned}$$

and

$$F_i^3(\mathbf{k}) + F_i F_r^2(\mathbf{k}) + c^2 F_i(\mathbf{k}) = \mathrm{Im}[S_1(\mathbf{k})S_2^*(\mathbf{k})S_3(\mathbf{k})]$$

where $F_r(\mathbf{k})$ and $F_i(\mathbf{k})$ are the real and imaginary components of the complex spectrum $F(\mathbf{k})$, respectively. It is then clear that we have two equations with three unknowns, i.e. $F_r(\mathbf{k})$, $F_i(\mathbf{k})$, and the real constant c (which is known only to Alice, being required for the first pass only). The equations therefore represent an under-determined system, and, in this context have infinitely many complex solutions (i.e. solutions in an algebraically closed field) or solutions that are inconsistent.

C. MATLAB Prototype Code for $\mathbf{r} \in \mathbb{R}^1$

Appendix C provides prototype software using MATLAB to implement the algorithm given in Section IX.A for one-dimensional array processing using function *TPP(key,step,c)*. This function, where *TPP* is an acronym for *Three-Pass Protocol*, has been designed to transfer an array of integers between two users (Alice and Bob, say) in the form of an ASCII delimited file which represents the initial plaintext - Plaintext.txt. The plaintext is typically a key which may be a concatenation of the keys used to execute the functions *POE* and *POD* given in Appendix B, for example. The function has three inputs:

- the *key* - a string of numbers between 0 and 9 - used by Alice for the first and third passes and a different *key* used by Bob for the second pass and the final decrypt;
- the *step* which has input values 1 (first pass), 2 (second pass), 3 (third pass) and 4 (for the final decrypt);
- the Spectral Embedding Constant c which is required for the first pass (for step=1) only, i.e. $c >> 1$ is required to be set by Alice to produce the first ciphertext; the value of this constant is not required to be known by Bob and is not used for any step other than step 1.

In each of the steps 1-3, the ciphertext is written to a file - 'Ciphertext.txt' - which is assumed to be sent (by email, for example) from Alice to Bob (step=1), from Bob back to Alice (step=2) and from Alice back to Bob (step=3). After step=3, Bob decrypts the ciphertext to recover the plaintext file which is output to the file 'Plaintext.txt'. In all cases, the MATLAB functions '*dmlread*' and '*dmlwrite*' are used for reading and writing the data to and from file, respectively.

By way of an example, and, in context of the operation of function *TPP* given in Appendix C, consider the exchange of a key given by the array 80 97 115 115 119 111 114 100 which is the ASCII decimal integer representation for the character string *Password*. It is assumed that Alice creates a file called 'Plaintext.txt' containing the integers given. Alice runs the function (for $opt = 1$) *TPP(1234,1,123456789)* where 1234 is Alice's key (known only to her), setting the value of c to 123456789 (also known only to Alice). Note that any value of c can be used by Alice up to the threshold associated with the precision available which needs to be quantified as in Figure 3; the precision is set to 32-bits in function *TPP*. After receipt of the file 'Ciphertext.txt' from Alice, Bob runs the function (for $opt = 2$) *TPP(4321,2)* where 4321 is Bob's key (known only to him) and sends the result - 'Ciphertext.txt' - back to Alice. After receipt of Bob's cipher, Alice runs the function *TPP(1234,3)* and sends the new ciphertext file - 'Ciphertext.txt' - back to Bob. Upon receipt of this file, Bob runs *TPP(4321,4)* to recover the key which is written in the output file 'Plaintext.txt'. Providing identical private keys are used by Alice and Bob for steps 1 and 3 and steps 2 and 4, respectively, the output file will contain the integer array 80 97 115 115 119 111 114 100. Thus Alice's key is passed to Bob using a TPP.

In step 1 of the function *TPP*, the input is zero padded, a zero being added to the first element of array. The reason for this is due to the re-normalisation condition which is applied in step 4 of the same function when the first element of the decrypt is eliminated from the output. In this way, re-normalisation becomes an intrinsic part of the process and is independent of the input and output plaintext's sent by Alice and received by Bob, respectively.

Given that the algorithms presented in Appendix B require three keys (for the Red, Green and Blue components), function *TPP* can be used to exchange these keys using a concatenated key

$$key\_R \parallel key\_G \parallel key\_B$$

where it is noted that this concatenation is for an array of integers not a string of numbers. The components of the array associated with each key are then concatenated into a string for application in functions *POE* and *POD*, the length of the integer strings representing each component key being limited to 10 digits (the limiting upper bound for the MATLAB random number generator used in these function). This approach can be extended to include an exchange of the Spectral Embedding Constant c when the option to apply WPOE is used in functions *POE* and *POD*. Alternatively the value of c may be exchanged separately using a second application of function *TPP*. Further, it is noted that function *TPP* can of course be used to pass any plaintext which is input as an ASCII decimal integer stream as given in the example above.

It is apparent that the implementation of phase-only encryption for a TPP key exchange may be subject to an attack through application of a phase retrieval algorithms as considered in Section VI.B. However, the practicality of implementing phase retrieval algorithms is dimension

dependent. It is well known that for $\mathbf{r} \in \mathbb{R}^2$, the phase retrieval problem has a range of solutions based on iterative numerical methods as briefly discussed and referenced in Section VI.B. However, for $\mathbf{r} \in \mathbb{R}^1$, the phase retrieval problem is ambiguousness, the determination of the phase within the extensive solution set being challenging and only able to be considered under suitable *a priori* assumptions or additional information. Thus an attack through application of a one-dimensional phase retrieval algorithm on any one of the passes (in particular, Steps 2 and 3 given in Section IX.A) can, at least for now, be assumed to be irrelevant. This statement should be appreciated within the context of possible future solutions to the one-dimensional phase retrieval problem. For example, it has recently been shown that a signal can be uniquely recovered from the Fourier amplitude alone if interference measurements between the unknown signal and a reference signal (unrelated to the unknown signal) are available [33], the investigation of this approach with regard to Section IX.B lying beyond the scope of this work. However, until one-dimensional phase retrieval solutions become as readily applicable and tractable as those developed for the two-dimensional case, the application of phase-only encryption for $\mathbf{r} \in \mathbb{R}^1$ and the key exchange solution compounded in function *TPP* will remain a significant challenge for cryptanalysts. In this sense, it is arguable that plaintext's in general should be transferred using variations on a theme of function *TPP* (to include generic data I/O, for example), including digital images, rather than using the function to exchange only the keys used for execution of the functions given in Appendix B, especially when conditioned according to the principles discussed in Section II.E.

X. Software Development and Usage

Appendix B and Appendix C are provided to give readers access to source code that implements the algorithms discussed in this paper using m-code. In both cases, copyright is attributed to J. M. Blackledge et al. and all rights are reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the organisation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The software listed in Appendices B and C is provided by the copyright holders and contributors *as is* and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright holders be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

XI. Conclusion

The material presented in this paper is predicated on proving the result that if the noise function associated with Equation (4) is considered to have a phase-only spectrum, then an exact and unique solution is available given by Equation (5). Further, if the phase function is taken to be a known stochastic function, then this result can be used to develop a phase-only encryption scheme as discussed in Section VII. The phase function can be generating using Equation (16) when the ciphertext is taken to be complex or Equation (17) when only the real component of the ciphertext is required. In this paper, the latter case has been considered.

As discussed in Section VI.B, in principle, an attack can be launched based on the application of a (two-dimensional) phase retrieval algorithm such as those used for X-ray crystallography. Thus to use such an encryption method in the field, the approach requires:

(i) application of a unique and cryptographically strong encryption algorithm(s) as discussed in Remark VII.3;
(ii) application of the same or a secondary cipher to prevent a correlation-based attack as compounded in Equation (13).

The theoretical foundations of phase-only deconvolution has, in this paper, been developed for n-dimensions. Thus the approach can be extended, without loss of generality, for n-dimensional data encryption. However, a MATLAB based application has been developed for the two-dimensional case, and, in particular for full colour images as discussed in Section VIII. It is noted that each colour component used in the algorithms provided in Appendix B can be replaced by an independent grey level or binary input (plaintext) image. The algorithms given rely on the encryption/decryption keys being known *a priori* and in this sense represent a standard symmetric encryption scheme. For this reason, the application of phase-only encryption to the key exchange problem has been considered as detailed in Section IX in which a three-pass protocol has been used. The algorithm presented in Appendix C can be used to exchange the keys required for implementation of algorithms given in Appendix B or can be used independently to exchange any plaintext composed of an array of integer numbers (assumed to be the decimal integers associated with the ASCII 7-bit code). Finally, it is noted that this approach can be 'coupled' with the method of information hiding developed by Blackledge et al [34] in which the functions *POE* and *POD* are used to transfer a covertext image.

Appendix A
Bayesian Cryptanalysis of Equation (4)

Given Equation (4), using Bayes theorem, an estimate for the plaintext $f(\mathbf{r})$ can be considered subject to the condition [1]

$$\frac{\partial}{\partial f}\ln \Pr(s \mid f) + \frac{\partial}{\partial f}\ln \Pr(f) = 0 \tag{24}$$

where $\Pr(s \mid f)$ denotes the *a posteriori* PDF of 's given f' and $\Pr(f)$ is the PDF of of the plaintext f. Clearly, in order to use this conditioning equation, models for the PDF's of n and f are required.

Suppose we consider $f(\mathbf{r})$ to be uniformly distributed and $n(\mathbf{r})$ to be Gaussian distributed. Then,

$$\frac{\partial}{\partial f} \ln \Pr(f) = 0$$

The *a posteriori* PDF $\Pr(s \mid f)$ is determined by the PDF of the cipher $\Pr(n)$ and from equation (4)

$$n(\mathbf{r}) = s(\mathbf{r}) - n(\mathbf{r}) \otimes f(\mathbf{r})$$

so that

$$\Pr(s \mid f) =$$

$$\frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{1}{2\sigma_n^2} \int d^n\mathbf{r} \mid s(\mathbf{r}) - n(\mathbf{r}) \otimes f(\mathbf{r}) \mid^2\right)$$

where σ_n is the standard deviation of $\Pr(n)$. We are then required to solve for $f(\mathbf{r})$ subject to the condition

$$\frac{\partial}{\partial f} \ln \Pr(s \mid f) = 0$$

Differentiating, application of the orthogonality principle [1] yields the equation

$$[s(\mathbf{r}) - n(\mathbf{r}) \otimes f(\mathbf{r})] \odot n^*(\mathbf{r}) = 0$$

In Fourier space (through application the convolution and correlation theorems) we have

$$S(\mathbf{k})N^*(\mathbf{k}) = \mid N(\mathbf{k}) \mid^2 F(\mathbf{k})$$

and hence, the Bayesian estimate for the Fourier transform of the plaintext is given by

$$F(\mathbf{k}) = \frac{S(\mathbf{k})N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}$$

APPENDIX B
PROTOTYPE MATLAB FUNCTIONS FOR PHASE-ONLY ENCRYPTION/DECRYPTION OF FULL COLOUR IMAGE

The functions given in this Appendix have not been exhaustively tested and are provided to give the reader a guide to the basic software engineering required to implement the computational procedures discussed in Section VII.A and VII.B, and, in turn, to help the reader appreciate the theoretical model developed in this paper. Where possible, the notation used for array variables and constants are based on the mathematical notation used in this paper. Note that the m-code given has been condensed spatially in order to conform to the format of this publication while minimising the number of pages required to present it. The software was developed and implemented using (64-bit) MATLAB R2018b.

A. Function to Phase-Only Encrypt a Colour Image

```
function []=POE(key_R,key_G,key_B,c,opt)
%FUNCTION: Phase-only Encryption (POE)
%          of a full colour image
%INPUTS:
%Plaintext colour image (Plaintext.bmp)
%Key_R: Key for Red component cipher;
%Key_G: Key for Green component cipher;
%Key_B: Key for Blue component cipher;
%c: Spectral Embedding Coefficient;
%OPTIONS:
%if opt==0, POE method is applied.
%if opt not==0, WPOE method is applied.
%OUTPUTS:
%Three .txt files of ciphertexts
%for RGB components.
%Read plaintext colour image (assumed
%to be a .bmp file) and show input image
%in figure using function 'imshow' (as
%required by the user).
I_C=imread('Plaintext.bmp');
figure(1), imshow(I_C);
%Extract RGB components of the input
%colour image, evaluate image size and
%convert RGB arrays to floating point
%form (double) - arrays taken to have
%N rows and M columns.
I_R =I_C(:, :, 1); I_G =I_C(:, :, 2);
I_B =I_C(:, :, 3); [N,M]=size(I_R);
I_R=im2double(I_R);  I_G=im2double(I_G);
I_B=im2double(I_B);
%Compute the following:
%(i) Uniformly distributed phase arrays
%for each RGB component using the input
%keys and MATLAB function 'rand'.
rng(key_R,'twister'); Theta_R=rand(N,M);
rng(key_G,'twister'); Theta_G=rand(N,M);
rng(key_B,'twister'); Theta_B=rand(N,M);
%(ii) The two-dimensional DFT of each
%array using function 'fft2', the phase
%angles (inverse tangent in radians
%between -pi and +pi) associated with
%the real and imaginary components of
%spectrum using function 'angle' and
%the phase-only spectra N_R, N_G & N_B.
N_R=exp(i*angle(fft2(Theta_R)));
N_G=exp(i*angle(fft2(Theta_G)));
N_B=exp(i*angle(fft2(Theta_B)));
%Encrypt input RGB components using phase
%only ciphers generated above by taking
%the DFT of the colour components, adding
%the results to the phase only cipher
%scaled by the embedding coefficient c
%(for opt==0) and take the real component
%of the inverse DFT (using the function
%'ifft2'), else, apply the WPOE method.
if opt==0
I_R=real(ifft2((fft2(I_R).*N_R)+c*N_R));
I_G=real(ifft2((fft2(I_G).*N_G)+c*N_G));
I_B=real(ifft2((fft2(I_B).*N_B)+c*N_B));
```

```
else
I_R=real(ifft2((fft2(I_R).*N_R) ...
+c*Theta_R.*N_R));
I_G=real(ifft2((fft2(I_G).*N_G) ...
+c*Theta_G.*N_G));
I_B=real(ifft2((fft2(I_B).*N_B) ...
+c*Theta_B.*N_B));
end
%Output the RGB components of the
%ciphertext (as floating point
%matrices with a precision set to
%32-bits) to three separate .txt files
dlmwrite('R_Enc.txt',I_R, ...
'delimiter',' ', 'precision',32);
dlmwrite('G_Enc.txt',I_G, ...
'delimiter',' ', 'precision',32);
dlmwrite('B_Enc.txt',I_B, ...
'delimiter',' ', 'precision',32);
%Concatenate the RGB component of the
%ciphertext and show normalised output
%ciphertext image in figure 2 using
%'imshow' (as required by user).
I_R=I_R./max(max(abs(I_R)));
I_G=I_G./max(max(abs(I_G)));
I_B=I_B./max(max(abs(I_B)));
I_C = cat(3,I_R,I_G,I_B);
figure(2), imshow(I_C);
```

B. Function to Phase-Only Decrypt a Colour Image

```
function []=POD(key_R,key_G,key_B,c,opt)
%FUNCTION: Phase-only Decryption (POD)
%          of a full colour image.
%INPUTS:
%Key_R: Key for Red component cipher;
%Key_G: Key for Green component cipher;
%Key_B: Key for Blue component cipher;
%c: Spectral Embedding Coefficient;
%OPTIONS:
%opt==0 POE method is applied
%opt not==0 WPOE method is applied.
%Note: if opt==0, the embedding
%coefficient is not used for
%decryption and any 'dummy value'
%may be passed (e.g. c=0), else, the
%value of c used to encrypt the image
%must be passed to obtain a decrypt.
%OUTPUT:
%Decrypt colour image (.bmp format).
%Read ciphertext of each RGB encrypted
%component from associated .txt files
%and compute array size
I_R=dlmread('R_Enc.txt');
I_G=dlmread('G_Enc.txt');
I_B=dlmread('B_Enc.txt');
[N,M]=size(I_R);
%Normalise RGB ciphertext components
%to recover colour ciphertext image I_C
%and display the result (as required).
I_R_N=I_R./max(max(abs(I_R)));
I_G_N=I_G./max(max(abs(I_G)));
I_B_N=I_B./max(max(abs(I_B)));
I_C = cat(3,I_R_N,I_G_N,I_B_N);
figure(1), imshow(I_C);
%Re-generate ciphers and phase only
%spectrum used to encrypt RGB
%components of the plaintext image
%using function POE.
rng(key_R,'twister'); Theta_R=rand(N,M);
N_R=exp(i*angle(fft2(Theta_R)));
rng(key_G,'twister'); Theta_G=rand(N,M);
N_G=exp(i*angle(fft2(Theta_G)));
rng(key_B,'twister'); Theta_B=rand(N,M);
N_B=exp(i*angle(fft2(Theta_B)));
%Decrypt the RGB components in Fourier
%space, apply inverse DFT (using 'ifft2'),
%take the real components output with
%undefined components being set to zero.
%If opt==0, decryption is undertaken
%without knowledge of c, else, weighted
%phase-only decryption is applied which
%requires c to be known.
if opt==0
I_R=real(ifft2((fft2(I_R).*conj(N_R))));
I_R(1,1)=0.0;
I_G=real(ifft2((fft2(I_G).*conj(N_G))));
I_G(1,1)=0.0;
I_B=real(ifft2((fft2(I_B).*conj(N_B))));
I_B(1,1)=0.0;
else
I_R=real(ifft2((fft2(I_R).*conj(N_R)) ...
-c*Theta_R)); I_R(1,1)=0.0;
I_G=real(ifft2((fft2(I_G).*conj(N_G)) ...
-c*Theta_G)); I_G(1,1)=0.0;
I_B=real(ifft2((fft2(I_B).*conj(N_B)) ...
-c*Theta_B)); I_B(1,1)=0.0;
end
%Concatenate the RGB components to
%reconstruct a colour image, write out
%decrypt to an image file (assuming .bmp
%format), and show image (as required).
I_C = cat(3,I_R,I_G,I_B);
imwrite(I_C,'Decrypt.bmp','bmp');
figure(2), imshow(I_C);
```

APPENDIX C
FUNCTION FOR IMPLEMENTATION OF A THREE-PASS PROTOCOL USING PHASE-ONLY ENCRYPTION

```
function []=TPP(key,step,c)
%FUNCTION: Exchange Plaintext composed
%of an array of integers between two
%user - User_1 and User_2 using the
%Three-pass Protocol (TPP) with
%Phase-only Encryption
%INPUTS:
%key: Key(s) used to execute TPP where
%'key' is a string of integer numbers
%between 0 and 9 with a maximum string
%length of 10 (the limiting upper bound
%for a MATLAB random number generator
%with a non-negative integer seed <2^32)
```

```
%step:
%step=1 - first pass (first encrypt)
%step=2 - second pass (second encrypt)
%step=3 - third pass (first decrypt)
%step=4 - decryption (second decrypt)
%c: Spectral Embedding Constant c>>1,
%user defined for the first pass only.
%Specification of c is not required
%for execution of steps 2, 3 and 4.
%Apply step 1 - first pass.
if step==1
%Read plaintext P (taken to be an
%array of  integers from 0 to 9)
%from an ASCII delimited file -
%Plaintext.txt - generated by
%User_1 to be transferred to User_2.
P=dlmread('Plaintext.txt');
%Zero pad the first element of
%array due to re-normalisation
%condition which needs to be
%applied in step 4 when the
%first element of the decrypt
%is eliminated from the output.
zero=zeros(1,1); P=[zero P];
N=size(P',1);%Compute size of P.
%Generate cipher using function
%'rand' seeded for first user
%defined key.
rng(key,'twister'); Theta=rand(1,N);
%Compute phase-only spectrum POS.
POS=exp(i*angle((fft(Theta))));
%Compute phase-only encrypted
%spectrum E, embed the result
%and return the real component
%of inverse DFT.
E=(fft(P).*POS)+c*POS;
E=real(ifft(E));
%Write out first pass ciphertext
%to file which is then sent by
%User_1 to User_2
dlmwrite('Ciphertext.txt',E, ...
'delimiter',' ','precision',32);
end
%Apply step 2 - second pass.
if step==2
%Read first passed ciphertext file
%(received by User_2 from User_1).
E=dlmread('Ciphertext.txt');
N=size(E',1); %Compute size of array.
%Computer fft of first pass ciphertext.
E=fft(E);
%Generate new cipher using function
%'rand' seeded by second user
%defined key.
rng(key,'twister'); Theta=rand(1,N);
%Compute phase-only encrypted
%spectrum and return real component
%of inverse DFT.
E=E.*exp(i*angle((fft(Theta))));
E=real(ifft(E));
%Write out second pass ciphertext
%to file which is then sent by
%User_2 to User_1.
dlmwrite('Ciphertext.txt',E, ...
'delimiter',' ','precision',32);
end
%Apply step 3 - third pass.
if step==3
%Read second passed ciphertext file.
%(received by User_1 from User_2).
E=dlmread('Ciphertext.txt');
N=size(E',1); %Compute size of array.
%Computer fft of second pass ciphertext.
E=fft(E);
%Generate cipher using function 'rand'
%seeded by first user defined key.
rng(key,'twister'); Theta=rand(1,N);
%Decrypt phase-only spectrum for
%first pass and return real
%component of inverse DFT.
E=E.*exp(-i*angle((fft(Theta))));
E=real(ifft(E));
%Write out third pass ciphertext
%to file which is then sent by
%User_1 to User_2.
dlmwrite('Ciphertext.txt',E, ...
'delimiter',' ','precision',32);
end
%Apply step 4 -
%Decryption of third pass cipher.
if step==4
%Read third pass cipher from file.
%(received by User_2 from User_1).
E=dlmread('Ciphertext.txt');
N=size(E',1); %Compute array size.
%Computer fft of second pass cipher.
E=fft(E);
%Generate cipher using function
%'rand' seeded by second user
%defined key.
rng(key,'twister'); Theta=rand(1,N);
%Decrypt phase-only spectrum for
%second pass, return real component
%of inverse DFT and re-normalise by
%setting the first element of the
%array to zero.
E=E.*exp(-i*angle((fft(Theta))));
P=real(ifft(E)); P(1)=0.0;
%Convert return to integer values,
%eliminate first element and square
%brackets associated with the array.
P=round(P); P(1)=[];
%Write out decrypt to
%Plaintext.txt file.
dlmwrite('Plaintext.txt',P, ...
'delimiter',' '); end
```

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Technological University Dublin, the University of Western Cape, the University of KwaZulu-Natal and the University of

Wales. Weston Govere and Dumisani Sibanda are sponsored by the Google Africa PhD Fellowship Program.

REFERENCES

[1] J. M. Blackledge, *Digital Signal Processing: Mathematical and Computational Methods, Software Development and Applications*, Edition 2, Woodhead Publishing: Series in Electronic and Optical Materials, 2006; eBook ISBN: 9780857099457. https://arrow.dit.ie/engschelebk/4/

[2] J. M. Blackledge, *Digital Image Processing: Mathematical and Computational Methods*, Woodhead Publishing Series in Electronic and Optical Materials, 2005; ISBN-13: 978-1898563495. https://arrow.dit.ie/engschelebk/3/

[3] P. C. Mogensen and J. Glc̈ksted, "Phase-only Optical Encryption", *Optics Letters*, vol. 25, issue 8, pp. 566-568, 2000.

[4] D. H. Seo and S. J. Kim, "Interferometric Phase-only Optical Encryption System that uses a Reference Wave", *Optics Letters*, vol. 28, issue 5, pp. 304-306, 2003.

[5] C. M. Shin and S. J. Kim, "Phase-Only Encryption and Single Path Decryption System Using Phase-Encoded Exclusive-OR Rules in Fourier Domain", *Optical Review*, vol. 13, no. 2, pp. 49-52, 2006.

[6] M. Z. He, L. Z. Cai, Q. Liu and X. L. Tang, "Phase-only Encryption and Watermarking based on Phase-shifting Interferometry", *Applied Optics*, vol. 44, issue 13, pp. 2600-2606, 2005.

[7] T. Li, Z. Miao and Y. Shi, "Ciphertext-Only Attack on Phase-Shifting Interferometry-Based Encryption", *IEEE Photonics Journal*, vol. 9, issue 5, 2017; https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8019774

[8] A. R. Al-Rawi, *Digital Rights Management using Steganocryptography*, *PhD Thesis*, Dublin Institute of Technology, 2013.

[9] J. M. Blackledge, S. Bezobrazov, P. Tobin and F. Zamora, "Cryptography using Evolutionary Computing", *Proc. IET ISSC2013*, Letterkenny, Co Donegal, Ireland, June 20-21, 2013.

[10] J Bernstein, J Buchmann and E Dahmen (Eds.), *Post-Quantum Cryptography, Springer*, 2009.

[11] Open Quantum Safe: Software for Prototyping Quantum-resistant Cryptography, 2016, https://openquantumsafe.org.

[12] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", *Proceedings of the 35th Annual IEEE Symposium of Foundations of Computers*, pp. 124-134, 1994.

[13] M. A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (10th Anniversary Edition), 2010; ISBN 978-1-107-00217-3. http://csis.pace.edu/ctappert/cs837-18spring/QC-textbook.pdf

[14] G Mogos, "Ciphertext-Policy Attribute-Based Encryption using Quantum Multilevel Secret Sharing Scheme", *IAENG International Journal of Computer Science*, vol. 45, no. 4, pp. 500-504, 2018. http://www.iaeng.org/IJCS/issues_v45/issue_4/IJCS_45_4_01.pdf

[15] G. H. Hardy, J. E. Littlewood and G. Pólya, *Inequalities*, Cambridge Mathematical Library, Edition 2, Cambridge University Press, 1952, ISBN: 0-521-35880-9; https://en.wikipedia.org/wiki/Minkowski_inequality

[16] W. H. Young, "On the Multiplication of Successions of Fourier Constants", *Proceedings of the Royal Society A*, vol. 87, pp. 331-339, 1912; https://en.wikipedia.org/wiki/Young\%27s_convolution_inequality

[17] P. Tobin, *On the Application of Pspice for Localized Cloud Security*, *PhD Thesis*, Dublin Institute of Technology, 2018; https://arrow.dit.ie/engdoc/111.

[18] J. M. Blackledge, *Cryptography Using Steganography: New Algorithms and Applications*, Lecture Notes, Centre for Advanced Studies, Warsaw University of Technology, Warsaw, 2012, ISBN: 978-83-61993-05-6;https://arrow.dit.ie/engscheleart2/40/

[19] J. M. Blackledge, "Information Hiding using Stochastic Diffusion for the Covert Transmission of Encrypted Images ", *Proc. of IET ISSC2010* UCC Cork, 23-24 June, 2010.

[20] J. M. Blacklegde, A Al-Rawi and R Hickson, "Multi-Channel Audio Information Hiding ", *Digital Audio Effects Conference (DAFx2012)*, York University, 17-21, September, pp. 309-316, 2012.

[21] J. M. Blackledge, P. Tobin, J. Myeza and C. M. Adolfo, "Information Hiding using Convolutional Encoding ", ISSC2018, *IEEE UK and Ireland Signal Processing Chapter*, Queens University Belfast, pp. 21-22 June 2018.

[22] H. E. Prabowo and T. Ahmad, "Improving Prediction Error Expansion based Data Hiding Method for Securing Confidential Data", *IAENG International Journal of Computer Science*, vol. 45, no. 4, pp. 540-551, 2018. http://www.iaeng.org/IJCS/issues_v45/issue_4/IJCS_45_4_06.pdf

[23] A. W. van der Vaart, *Asymptotic Statistics*, Cambridge University Press, 1998; ISBN 978-0-521-49603-2.

[24] D. Freedman and D. Lane, "The Empirical Distribution of Fourier Coefficients ", *The Annals of Statistics*, vol. 8, no. 6, pp. 1244-1251, 1980.

[25] J. M. Blackledge, P. Tobin and S. Bezobrazov, "Cryptography using Artificial Intelligence", *The International Joint Conference on Neural Networks (IJCNN2015)*, Killarney, Ireland, 12-17 July, 2015.

[26] W. Diffie and M. E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976; https://ee.stanford.edu/~hellman/publications/24.pdf

[27] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", *Communications of the Association for Computing Machinery (ACM)*, vol. 21, no. 2, pp. 120-126, 1978.

[28] J. R. Fienup, "Phase Retrieval Algorithms: A Comparison", *Applied Optics*, vol. 21, no. 15, pp. 2758-2769, 1982; https://www.osapublishing.org/DirectPDFAccess/7504D185-F6FF-F581-9D5EFBBF8D755809_26002/ao-21-15-2758.pdf?da=1&id=26002&seq=0&mobile=no

[29] J. R. Fienup, "Phase Retrieval Algorithms: A Personal Tour", *Applied Optics*, vol. 52, no. 1, pp. 45-56, 2013; http://www2.optics.rochester.edu/workgroups/fienup/PUBLICATIONS/JRF_PR-Tour_AO2013.pdf

[30] A. Menezes, P. Van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp. 500-642, 1996; ISBN: 0-8493-8523-7.

[31] J. L. Massey and J. K. Omura *Method and Apparatus for Maintaining the Privacy of Digital Messages Conveyed by Public Transmission*, US Patent US4567600A, 1986. https://patents.google.com/patent/US4567600

[32] K. Sakurai and H. Shizuya, "A Structural Comparison of the Computational Difficulty of Breaking Discrete Log Cryptosystems", *Journal of Cryptology*, vol. 11, pp. 29-43, 1998.

[33] R. Beinert, "One-dimensional Phase Retrieval with Additional Interference Measurements", *Cornell University, arXiv:1604.04489v1*, 2016. https://arxiv.org/pdf/1604.04489.pdf

[34] J. M. Blackledge, P. Tobin, J. Myeza and C. M. Adolfo, "Information Hiding with Data Diffusion using Convolutional Encoding for Superencryption", *International Journal for Pure and Applied Mathematics*, vol. 7, no. 4, pp. 319-356, 2017; https://arrow.dit.ie/cgi/viewcontent.cgi?article=1149&context=engscheleart2

Jonathan Blackledge holds a BSc and PhD in Physics from London University and a Doctorate in Mathematical Information Technology from the University of Jyvaskyla, Finland. He has published over 300 scientific and engineering research papers and reports including 14 teaching and research books, has filed 20 patents and technologies to license and has been supervisor to over 300 MSc/MPhil and 67 PhD research graduates. After a career in both academia and industry, in 2008 he was elected to become the Science Foundation Ireland's Stokes Professor at the Technological University Dublin where he has been an Honorary Professor since 2012. He holds Visiting Professorships with the University of Western Cape, Warsaw University of Technology and the University of Wales and Fellowships with a number of UK Institutes and Societies including the Institute of Physics, the Institute of Mathematics and its Applications, the British Computer Society and the Institution of Engineering and Technology. He is currently directing research programs for the defence sector focusing on post-quantum cryptology. Weston Govere has a BSc Honours Degree in Mathematics from Midlands State University, Zimbabwe and a MSc in Information Theory, Coding and Cryptography from Mzuzu University, Malawi. He is currently a Lecturer in the Department of Information Security and Assurance at the Harare Institute of Technology and a research associate registered with the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, South Africa. Dumasani Sibanda has a BSc Honours Degree in Statistics from the University of Zimbabwe and a MSc in Information Theory, Coding and Cryptography from Mzuzu University, Malawi. He is currently a Lecturer at the Harare Institute of Technology and a research associate in the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, South Africa.