2019

Phase-only Digital Encryption using a Three-pass Protocol

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

Paul Tobin
Technological University Dublin, paul.tobin@tudublin.ie

W. Govere
University of KwaZulu-Natal, South Africa., wgovereh@gmail.com

See next page for additional authors

Recommended Citation

Blackledge, J. (2019) Phase-only Digital Encryption using a Three-pass Protocol, *30th Irish Signals and Systems Conference (ISSC 2019)* Mon 17th - Tue 18th June 2019, Maynooth University, Ireland. doi.org/10.21427/am23-g891

Authors

Jonathan Blackledge, Paul Tobin, W. Govere, S. Sibanda, and C. M. Adolfo

Phase-only Digital Encryption using a Three-pass Protocol

J. M. Blackledge
Stokes Professor, Science Foundation Ireland.
Visiting Professor, Wrexham Glyndwr University, UK.
Professor Extraordinaire, University of Western Cape, SA.
jonathan.blackledge@dit.ie

P. Tobin
Department of Electrical and Electronic Engineering,
Technological University Dublin,
Republic of Ireland.
paul.tobin@dit.ie

W. Govere and S. Sibanda
School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal, South Africa.
wgovereh@gmail.com and dumisanisibanda@yahoo.co.uk

C. M. Adolfo
Department of Electrical and Electronic Engineering,
Technological University Dublin, Ireland.
cidmathew.adolfo1@gmail.com

***Abstract*—This paper considers an application of phase-only digital encryption to the three-pass protocol leading to a new 'no-key-exchange algorithm'. After providing a study on the theoretical background to the method, an algorithm is presented on a step-by-step basis together with three examples of cryptanalysis. A prototype MATLAB function is provided for validation of the approach and for further development by interested readers.**

***Index Terms*—Convolution, deconvolution, phase-only spectrum, stochastic phase function, cryptanalysis, three-way pass protocol.**

I. Introduction

An important theme in the development of any encryption algorithm, at least, within the context of a given encryption model, is the 'diffusion' and 'confusion' that occurs in the transformation of an (input) plaintext to an (output) ciphertext. Confusion refers to condition that the correlation between the key and the ciphertext is as complex and as intricate as possible. Diffusion refers to the property that the statistical distribution of the plaintext is dissipated in the distribution of the ciphertext so that the statistical signature of the plaintext is not present in the ciphertext [1]. Ideally, what is required is a process that outputs a uniformly distributed ciphertext. Encrypting data using a known cipher (with a uniform distribution) and a specific encryption process does not always guarantee an output ciphertext with a uniform distribution. However, what matters most, is that the distribution of the plaintext is dissipated effectively over the full extent of the plaintext. In this context, we consider the following encryption model:

$$s(x)=\overbrace{\underbrace{n(x)\quad\otimes\quad f(x)}_{\text{Stochastic Diffusion}}\;+\;cn(x)}^{\text{Stochastic Confusion}} \tag{1}$$

where the functions $f(x)$ - the information function, $n(x)$ - the noise function and $s(x)$ - the signal, denote the the plaintext, the cipher and the ciphertext, respectively. The operator $\otimes$ denotes the convolution integral and c is a constant. We refer to the convolution operation $n(x)\otimes f(x)$ as the process of (stochastic) diffusion and the addition of term $cn(x)$ as the process of (stochastic) confusion as illustrated in Equation (1), both terms on the RHS of Equation (1) being stochastic functions.

Stochastic diffusion is 'maximized' by ensuring that $n(x)$ is, ideally, uniformly distributed. Maximum confusion is then determined by the extent to which $n(x)$ dominates $s(x)$ under the condition that $\|n(x)\otimes f(x)\| << \|cn(x)\|$ which is clearly determined by the value of c. This ensures that the statistical signature of the ciphertext is determined by the cipher alone.

Judging from the open literature, and, to the best of the author's knowledge, the application of Equation 1 using a phase-only cipher $n(x)$ and its application to a three-pass protocol is a new and original contribution to the field, given that phase-only encryption methods have traditionally been a prerogative of optical cryptography [2].

II. Deconvolution for Phase-only Functions

Given Equation (1), it is clear that in order to decrypt the signal $s(x)$ to recover the information $f(x)$, it is necessary to deconvolve $s(x)$ given the cipher $n(x)$. Application of a phase-only cipher provides an exact and unique solution to this problem. This is compounded in the following theorem.

Theorem 2.1: If $n(x)$ has a phase-only spectrum, then the deconvolution problem associated with Equation (1) is well-posed, i.e. $f(x)$ can be recovered from $s(x)$ exactly and uniquely, except for the value of $f(x)$ at $\mid x\mid=0$ which remains undefined.

Proof 2.1: Let $\Theta(k)$ be the phase function of a unit amplitude phase-only spectrum such that

$$\exp[i\Theta(k)]=N(k)\leftrightarrow n(x)$$

where $\leftrightarrow$ denotes transformation to Fourier or k-space. Applying the convolution theorem to Equation (1), we can write

$$S(k)=\exp[i\Theta(k)]F(k)+c\exp[i\Theta(k)]$$

978-1-5386-6046-/18/$31.00 © 2019 IEEE

where $S(k) \leftrightarrow s(x)$ and $F(k) \leftrightarrow f(x)$ and it is then clear that

$$\exp[-i\Theta(k)]S(k) = F(k) + c$$

Hence, using the correlation theorem (where $\odot$ denotes the correlation integral)

$$f(x) + c\delta(x) = n^*(x) \odot s(x)$$

where $\delta(x)$ is the Dirac delta function. However, since the function $c\delta(x)$ only affects the value of $f(x)$ at $\mid x \mid= 0$ and has no influence for all values of $\mid x \mid> 0$, we can write

$$f(x) = n^*(x) \odot s(x), \ f(x = 0) = 0 \tag{2}$$

which provides the solution for $f(x)$ subject to what we call the 're-normalization condition'.

Corollary 2.1: If, for any norm, we consider the ratio $R = \|n(x) \otimes f(x)\|/\|cn(x)\|$, then it is clear that $R \geq 0$ is a measure of the Signal-to-Noise Ratio (SNR) since $\|s(x)\| \leq \|n(x) \otimes f(x)\| + \|cn(x)\|$ leading to the inequality $\|s(x)\|/\|cn(x)\| \leq 1 + R$. Hence, if we define the SNR as $1 + R$, then Equation (2) shows that $f(x)$, $\forall \mid x \mid> 0$, can be uniquely recovered from $s(x)$ whatever the magnitude of the SNR, i.e. however small the value of the SNR or equivalently, however large the value of c becomes.

Remark 2.1: Theorem 2.1 and Corollary 2.1 are theoretical results associated with the use of piecewise continuous functions. Thus, the statement 'however large the value of c', is *null and void* for applications involving numerical operations on finite discrete arrays when the value of c becomes subject to an upper bound that depends on the floating point precision applied. In other words, the numerical value of c that is applied places limits on the accuracy to which a discretized form of $f(x)$ can be deconvolved from a discretized form of $s(x)$. A detailed quantification of this numerical issue lies beyond the scope of the paper but in general, for $c = 10^m$, an m-bit precision is required.

Remark 2.2: The results compounded in Theorem 2.1 and Corollary 2.1 are of no intrinsic value to the deconvolution problem in general which occurs in the applications of digital signal processing, for example. This is because it can rarely, if ever, be assumed that an Impulse Response Function is characterized by a deterministic or stochastic phase-only spectrum. Further, natural noise can not, in general, be assumed to be characterized by phase-only functions. Thus, it should be understood that Theorem 2.1 is strictly only applicable to the convolution model given in Equation (1) when $n(x)$ has a phase-only spectrum and thereby, has no applicability to signal processing in general. However, as explored in this paper, Theorem 2.1 does have applications in the area of cryptography. This is because the diffused plaintext $n(x) \otimes f(x)$ can be completely embedded in the (phase-only) cipher $cn(x)$ for $c >> 1$, thereby fully dissipating the statistical signature of the plaintext in the distribution of the cipher. This maximizes the 'confusion' of the ciphertext within the context of the encryption model given by Equation (1).

III. Encryption using Phase-Only Stochastic Functions

Consider the encryption model given by Equation (1) where

$$n(x) \leftrightarrow \ \exp[i\Theta(k)], \ \Theta(k) \in [-\pi, \pi]$$

The function $n(x)$ is a cipher generated by some key dependent algorithm characterized by a phase-only spectrum with a random phase function $\Theta(k)$. For a plaintext function $f(x)$ that is real, the ciphertext is taken to be given by $\text{Re}[s(x)]$ and has some Probability Density Function (PDF) denoted by $\Pr\{\text{Re}[s(x)]\}$.

In Fourier space, Equation (1) now becomes

$$S(k) = F(k)\exp[i\Theta(k)] + c\exp[i\Theta(k)] \tag{3}$$

and it is clear that the value of c controls the magnitude of the term $\exp[i\Theta(k)]$ compared with the term $F(k)\exp[i\Theta(k)]$. As c increases in magnitude, the spectrum $S(k)$ becomes dominated by the phase-only spectrum $\exp[i(\Theta(k)]$. Thus by choosing a large value for c we can 'embed' the spectrum $F(k)\exp[i\Theta(k)]$ in the phase-only spectrum $\exp[i\Theta(k)]$ in the knowledge that $F(k)$ can be recovered exactly as a consequence of Theorem 2.1. For this reason, we refer to c as the Spectral Embedding Coefficient (SEC).

This result is invariant of the phase function $\Theta(k)$ that is chosen for $c >> 1$. The output will be dominated by the stochastic behaviour associated with the PDF of $n(x)$ thereby eliminating the statistical signature associated with the term $n(x) \otimes f(x)$. Hence, the distribution of the plaintext is dissipated in the distribution of the cipher through both convolution (diffusion) and addition (confusion) but where the addition of the cipher is the dominating effect for $c >> 1$, subject to the floating point accuracy required to give a decrypt with an acceptable accuracy, as discussed in Remark 2.1. It is this observation that is fundamental to using Theorem 2.1 for encryption.

Let the stochastic phase function $\Theta(k)$ be conditioned to have a uniform distribution and wrapped between $-\pi$ and π radians so that we can write

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

There are two principal ways to compute the random phase function $\Theta(k) \in [-\pi, \pi] \ \forall k$: (i) Generate a normalized uniformly distributed function $R(k) \in [0, 1]$, say, and let

$$\Theta(k) = 2\pi \left[R(k) - \frac{1}{2}\right] \tag{4}$$

(ii) Construct $\Theta(k)$ by taking the Fourier transform of a random (uniformly distributed) variable $r(x) \in [0, 1]$, say, i.e.

$$\Theta(k) = \text{atan2}[R(k)], \ \ R(k) \leftrightarrow r(x) \tag{5}$$

In this case, atan2 is taken to yield 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 $r(x)$ is real and has no symmetry, $R(k)$ has a symmetric real component and an asymmetric imaginary component. The 4-quadrant phase function is therefore an asymmetric function, i.e. $\Theta(k) = -\Theta(-k)$. Thus although $\Pr\{\mid \Theta(k) \mid\}$ is uniformly distributed, the stochastic signature of $\mid \Theta(k) \mid$ in one half-space is repeated in the other. The two half-spaces are therefore not statistically uncorrelated.

The principal differences between the two approaches for computing the phase function given above is that application of Equation (4) requires the complex ciphertext to be used in order to recover the plaintext, whereas Equation (5) requires only the real component of the ciphertext to be retained. Application of Equation (5) therefore reduces the data storage/transmission requirements needed to recover the plaintext by 50% albeit at the 'price' of using a stochastic phase function with the property $\Theta(k) = -\Theta(-k)$ and the computational overhead of computing the spectrum $R(k)$ from $r(x)$ using, in practice, a Fast Fourier Transform. For this reason, Equation (5) is used to compute the stochastic phase-only spectrum.

IV. Application using a Three-pass Protocol

The principle of the Three-pass Protocol is well known as are the algorithms that have been developed for its implementation. These include the Shamir three-pass protocol [3] and the Massey-Omura method [4]. 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 but a new private key K_B known only to Bob and sends the now double encrypted plaintext back to Alice. Upon receipt, and critically, assuming the encryption algorithm is commutative, Alice can decrypt the doubly 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 a 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: (i) The encryption algorithm used must be commutative and strong enough so that the ciphertext cannot be broken using a known algorithm attack based on an intercept of any pass, particularly the single encrypted first and third passes; (ii) the keys used must be of a sufficient length to make an exhaustive attack impracticable on any pass; (iii) if the encrypted information is intercepted for each of the three passes, it is not possible to determine the plaintext from the three intercepts (assumed to be complete intercepts in each case). 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 some 'computational difficulty' in this respect. 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 [5]. In this section we consider an application of the three-pass protocol using a phase-only encryption (commutative) algorithm. As discussed further in Section IV, the 'computational difficulties' of breaking the ciphertext in this case are compounded in: (i) a lack of knowledge on the exact value of the SEC that is used by Alice in the first pass (and only the first pass); (ii) the inability to solve the one-dimensional phase retrieval problem due to the Fundamental Theorem of Algebra; (iii) the inability to uniquely solve an under-determined cubic polynomial using a three-intercept cryptanalysis.

A. Basic Algorithm

Consider the case when Alice wishes to exchange a single plaintext given by the real function $f(x) \leftrightarrow F(k)$, the plaintext for $f(x = 0)$ being taken to be redundant due to the re-normalization condition associated with Equation (2). Alice generates the random phase cipher $\Theta_1(k)$ and similarly, Bob generates random phase cipher $\Theta_2(k)$. These phase functions are computed through application of the Fourier transform method compounded in Equation (5). The algorithm(s) for generating $r(x)$ from which these phase functions are generated, are taken to be cryptographically strong and ideally personal to Alice and Bob through application of an evolutionary computing approach [6] including the keys used to seed them. The following steps are then applied.

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

$$S_1(k) = F(k)\exp[i\Theta_1(k)] + c\exp[i\Theta_1(k)] \tag{6}$$

and sends $\text{Re}[s_1(x)]$ of $s_1(x) \leftrightarrow S_1(k)$ to Bob.

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

$$S_2(k) = S_1(k)\exp[i\Theta_2(k)] \tag{7}$$

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

Step 3: Alice decrypts Bobs ciphertext $S_2(k) \leftrightarrow \text{Re}[s_2(x)]$ using the equation

$$\begin{aligned} S_3(k) = S_2(k)\exp[-i\Theta_1(k)] &= S_1\exp[i\Theta_2(k)]\exp[-i\Theta_1(k)] \\ &= [F(k)+c]\exp[i\Theta_1(k)]\exp[-i\Theta_1(k)]\exp[i\Theta_2(k)] \\ &= [F(k)+c]\exp[i\Theta_2(k)] \end{aligned} \tag{8}$$

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

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

$$F(k) + c = S_3(k)\exp[-i\Theta_2(k)] \tag{9}$$

The plaintext is then given by $\text{Re}[f(x)] \quad \mid x \mid > 0$ where $f(x) \leftrightarrow F(k)$ given that $\text{Re}[f(0)]$ is undefined, or, with application of the re-normalization condition $\text{Re}[f(0)] = 0$.

Note that the value of c used in the first pass can be randomly generated once the numerical upper bound of this constant has been established subject to the floating point accuracy used to undertake Steps 1 - 4.

B. Three-intercept Cryptanalysis

Assume that an attack is launched to estimate $f(x)$ based on knowledge of the three-pass protocol given in Section IV.A and accurate records of the functions $S_1(k)$, $S_2(k)$ and $S_3(k)$ obtained by intercepting the transmission associated with Steps 1-3 (by taking the Fourier transform of the results). Given Equations (6) - (9), we can then eliminate the ciphers Θ_1 and Θ_2 to obtain the equation

$$F(k)+c=\frac{S_1(k)S_2^*(k)S_3(k)}{\mid S_1(k)\mid^2}=\frac{S_1(k)S_2^*(k)S_3(k)}{\mid F(k)\mid^2+c^2},\ c\to\infty \tag{10}$$

and it is clear that to obtain $F(k)$ we are required to solve a cubic equation for an unknown value of $c >> 1$ (which is known only to Alice, being required for the first pass only). Equation (10) is under-determined, and, in this context, has infinitely many complex solutions that are inconsistent (solutions in an algebraically closed field). Thus a unique solution for $f(x)$ given knowledge of $S_1(k)$, $S_2(k)$ and $S_3(k)$ is not possible.

C. Bayesian Cryptanalysis

Bayesian analysis can be used to generate a Maximum Likelihood estimate for $f(x)$ given Equation (1). This requires a model for the statistical distribution of $n(x)$ to be established. If we consider the case when $n(x)$ is characterized by a Gaussian PDF, the estimate for $f(x)$, $\hat{f}(x)$ say, is given by [7], with $G(k)\leftrightarrow g(x)$,

$$\hat{f}(x)=g^*(x)\odot s(x)\ \text{ where }\ G^*(k)=\exp[i\Theta(k)]$$

In regard to phase-only encryption, a known algorithm attack is required to focus on the phase function $\Theta(k)$. This rules out the ability to develop a useful Bayesian attack because: (i) the statistical signature of the ciphertext is dominated by the function $\text{Re}[n(x)]$ (providing $c >> 1$) and the statistical signature of the plaintext $f(x)$, after diffusion with $n(x)$, is therefore not available in practice; (ii) even though the distribution of $\text{Re}[n(x)]$ is known, the phase function $\Theta(k)$ is uniformly distributed and Bayesian estimation is not possible for uniformly distributed variables.

D. Correlation Cryptanalysis

Consider the case when the ciphertext in Step 1 is intercepted. From Equation (6) we can construct the power spectrum

$$\begin{aligned}\mid S_1(k)\mid^2 &=\mid F(k)\exp[i\Theta_1(k)]+c\exp[i\Theta_1(k)]\mid^2\\ &=\mid F(k)\mid^2+c^2\left[1+\frac{2}{c}\text{Re}[F(k)]\right]\end{aligned}$$

For $c >> 1$ it is then clear that (in the asymptotic case)

$$\mid S_1(k)\mid^2=\mid F(k)\mid^2+c^2,\ c\to\infty$$

or, using the correlation theorem, where $S_1(k)\leftrightarrow s_1(x)$,

$$\begin{aligned}s_1^*(x)\odot s_1(x)&=f^*(x)\odot f(x)+c^2\delta(x)\\ &=f^*(x)\odot f(x);\ \mid x\mid>0,\ c\to\infty\end{aligned}$$

Autocorrelating the ciphertext therefore leads to a decryption problem that is equivalent to the phase retrieval problem, i.e. given that $f(x)\leftrightarrow F(k)=\mid F(k)\mid\exp[i\Theta(k)]$, then if only $\mid F(k)\mid$ is known, we are required to estimate the phase function $\Theta(k)$ upon which $f(x)$ 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. Moreover, the practicality of implementing phase retrieval algorithms is dependent on the dimension. It is well known that for the two-dimensional case, phase estimation algorithms have been developed to provide approximate solutions, especially in regard to X-ray crystallography, for example, where the intensity of an X-ray diffraction pattern in the far-field is determined by the two-dimensional Fourier transform of the diffracting object (i.e. the crystal) [8]. However, for the one-dimensional case considered here, 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. This is a consequence of the Fundamental Theorem of Algebra which states that every single-variable polynomial with complex coefficients has at least one complex root. This theorem fails for polynomials of two variables and it is the inability to factor polynomials of two variables that makes the two-dimensional phase retrieval possible and prevents the one-dimensional phase retrieval problem from being solved. This is because the ability to factor polynomials generates ambiguities where multiple spectral phases correspond to the same data. Thus any attack associated with attempting to solve the one-dimensional phase retrieval problem applied to any one or all of the three passes can be assumed to fail and the application of phase-only encryption considered here will therefore remain a significant challenge for a cryptanalyst. This statement should of course 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 [9].

It may also be possible to apply a variation on the theme of the two-to-one dimensional processing equivalence principle via application of the Radon and inverse Radon transforms ($\hat{R}$ and $\hat{R}^{-1}$, respectively) first considered in [10] and compounded in the equation $\hat{P}_2f(x,y)\equiv\hat{R}^{-1}\hat{P}_1\hat{R}f(x,y)$ where $\hat{P}_1$ denotes the one-dimensional process and $\hat{P}_2$ denotes the equivalent two-dimensional process (which may or may not be directly applicable as a two-dimensional processing algorithm). In this context, an attempt at solving the one-dimensional phase retrieval problem could be considered using a one-to-two dimensional processing equivalence principle based on the equation $\hat{P}_1 f(x) \equiv \hat{R}\hat{P}_2\hat{R}^{-1}f(x)$, noting that $\hat{R}^{-1}f(x)$ is a symmetric image to which the two-dimensional phase retrieval process denoted by $\hat{P}_2$ is then applied.

E. Prototype MATLAB Function

Appendix A provides prototype software using MATLAB to implement the algorithm given in Section IV.A, compounded in function TPP, an acronym for Three-Pass Protocol. The function has been designed to transfer a .txt file between two users (Alice and Bob) and has four inputs: (i) the *key* - a string of numbers between 0 and 9 (consisting of a maximum of 10 digits which is the limiting upper bound for a MATLAB random number generator with a non-negative integer seed $< 2^{32}$) - used by Alice for the first and third passes and a different *key* used by Bob for the second pass and the final decrypt; (ii) the *step* which is assigned input values 1 (first pass), 2 (second pass), 3 (third pass) and 4 (for the final decrypt); (iii) a user-specified *filename*; (iv) the Spectral Embedding Constant c which is required for the first pass (*step*=1) only. In each of the 1-3, the ciphertext is written to (and read from for *step*=2 and *step*=3) a file with default filename ‘Ciphertext.txt’. This numerical data 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 which is output to a user specified file. For each step, the MATLAB functions *dmlwrite* and/or *dmlread* are used for writing and/or reading the ciphertext to a file, respectively. For *step*=1, the input to function TPP (after conversion to an integer array) is zero padded, a zero being added to the first element of the array. The reason for this is due to the re-normalization condition which is applied in *step*=4 of the same function when the first element of the decrypt is eliminated from the output, thereby making the re-normalization process an intrinsic feature of the function and independent of the input/output data.

A simple graphical example of the data associated with function TPP is given in Figure 1 which shows line plots of an input plaintext (a single short sentence consisting of 114 ASCII characters) and output ciphertext's associated with the three passes (steps 1-3) for $c = 1234.56789$ using the keys 1234 and 4321 for *step*=1, *step*=3 and *step*=2, *step*=4, respectively. These results are associated with running the function TPP in the following sequence: *TPP(1234,1,'Plaintext.txt',1234.56789), TPP(4321,2), TPP(1234,3), TPP(4321,4,'Decrypt.txt')* where Plaintext.txt is the input plaintext file (created by Alice) and Decrypt.txt is the final decrypt (generated by Bob with *step*=4). Each Ciphertext.txt file consists of an array of floating point numbers with 32 digits whose numerical scale is determined by (but not directly related to) the magnitude of c. It is noted that none of the three ciphers correlate with each other but that each cipher has the same statistical distribution and therefore do correlate statistically with each other.

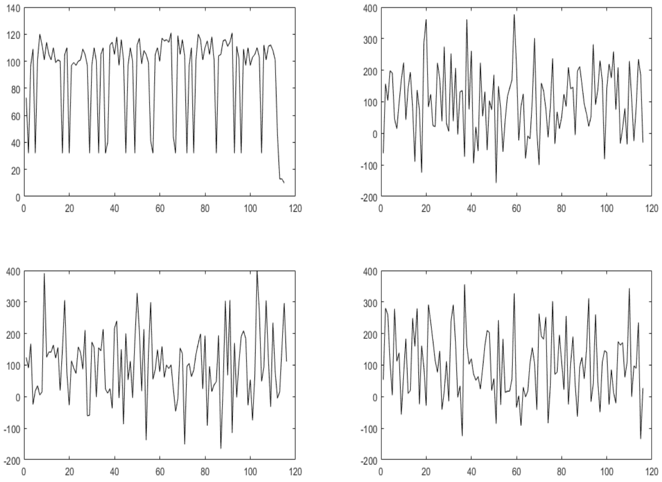

Fig. 1. Example plots of plaintext (top-left), first pass ciphertext (bottom-left), second pass ciphertext (top-right) and third pass ciphertext (bottom-right).

V. Discussion and Conclusions

The material presented in this paper is predicated on proving the result that if the noise function $n(x)$ given in Equation (1) is considered to have the same phase-only spectrum, then an exact deconvolution is available given by Equation (2). This result has been applied to the exchange of plaintext using a three-pass protocol without the need to exchange any keys.

Cryptanalysis shows that if the ciphertext is intercepted in each pass, the arbitrary value of the Spectral Embedding Coefficient used in the first pass (which can be many tens of orders of magnitude depending on the floating point precision available) yields a third order polynomial equation that is under-determined, i.e. an equation which has no solution. Bayesian and correlation attacks have been shown to be invalid, in the latter case, because of the lack of any available solution to the one-dimensional phase retrieval problem; a consequence of the Fundamental Theorem of Algebra. In this sense, the method cannot be broken subject, of course, to the possibility of future exceptions. The algorithm presented in Appendix A - function TPP - can therefore be used to exchange any ACSII text data associated with a user-specified plaintext file. The function can and should be modified and extended by interested readers to encrypt different data types as required, the value of c used in the first pass being automated using, for example, the code

```
c=rand(1,1)*10^(round(rand(1,1)*6))
```

which assigns a real random floating point value between 0 and 10^6 to the SEC.

Appendix A
MATLAB Function for the Implementation of a Three-pass Protocol using Phase-only Encryption

The function given in this Appendix has not been exhaustively tested and has no data error checks. It is provided to give the reader a guide to the basic programming required to implement the computational procedures discussed in Section IV.A. The code given has been commented but is highly condensed in order to comply with the prescribed page limit for this publication.

```
function []=TPP(key,step,filename,c)
%FUNCTION: Exchange plaintext file.
%INPUTS: key-an integer string (0-9)
%with maximum string length of 10 digits.
%step=1-first pass to encrypt plaintext.
%step=2-second pass, encrypt ciphertext.
%step=3-third pass giving first decrypt.
%step=4-output decrypt writing data to
%a user specified file.
%c>>1-constant (for first pass only).
if step==1 %Read plaintext P from file.
fid = fopen(filename,’r’);
P=fscanf(fid, ’%c’); fclose(fid);
P=round(P); %Convert to integers.
%Zero pad the first element of array.
zero=zeros(1,1); P=[zero P];
N=size(P’,1); %Compute size of P.
%Generate cipher using function ’rand’
%seeded by key specified by first user.
rng(key,’twister’); Theta=rand(1,N);
%Compute Phase-Only Spectrum (POS).
POS=exp(i*angle((fft(Theta))));
%Compute phase-only encrypted spectrum
%E and return the real part of ifft.
E=(fft(P).*POS)+c*POS; E=real(ifft(E));
%Write out first pass ciphertext to file
dlmwrite(’Ciphertext.txt’,E, ...
’delimiter’,’ ’,’precision’,32); end
%Processing for step 2 (second pass).
if step==2 %Read first pass ciphertext.
E=dlmread(’Ciphertext.txt’); N=size(E’,1);
%Computer fft of first pass ciphertext
E=fft(E); %and generate new cipher using
%key specified by the second user.
rng(key,’twister’); Theta=rand(1,N);
%Compute phase-only encrypted spectrum,
%return the real component of ifft
E=E.*exp(i*angle((fft(Theta))));
E=real(ifft(E)); %and write data to file.
dlmwrite(’Ciphertext.txt’,E, ...
’delimiter’,’ ’,’precision’,32); end
%Apply processing for step 3 - third pass.
if step==3 %Read second pass ciphertext.
E=dlmread(’Ciphertext.txt’); N=size(E’,1);
%Computer fft of second pass ciphertext
E=fft(E); %and generate cipher using key
%specified by first user.
rng(key,’twister’); Theta=rand(1,N);
%Decrypt phase-only spectrum for first
%pass, return real component of ifft
E=E.*exp(-i*angle((fft(Theta))));
E=real(ifft(E)); %and write data to file.
dlmwrite(’Ciphertext.txt’,E, ...
’delimiter’,’ ’,’precision’,32); end
%Apply processing for step 4 -
%Decryption of third pass cipher.
if step==4 %Read third pass cipher.
E=dlmread(’Ciphertext.txt’); N=size(E’,1);
%Compute fft of second pass cipher
E=fft(E); %and generate cipher seeded
%by key specified by the second user.
rng(key,’twister’); Theta=rand(1,N);
%Decrypt phase-only spectrum, return real
%component of ifft and re-normalize by
%setting first element of the array to 0.
E=E.*exp(-i*angle((fft(Theta))));
P=real(ifft(E)); P(1)=0.0;
%Convert return to integer values,
%and eliminate first element.
P=round(P); P(1)=[];
%Write out decrypt to filename.txt
fid = fopen(filename,’wt’);
fprintf(fid, ’%c’, P); fclose(fid); end
```

Acknowledgment

The authors would like to acknowledge the support of the Science Foundation Ireland, Technological University Dublin, Ireland, University of Western Cape, South Africa, University of KwaZulu-Natal, South Africa, Wrexham Glyndwr University, UK and Secure Intelligence Limited, UK.

References

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

[2] P. C. Mogensen and J. Glückstad, *Phase-only Optical Encryption*, Optics Letters, vol. 15, no. 25(8), pp. 566-568, 2000.

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

[4] 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

[5] 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.

[6] 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.

[7] 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/

[8] 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/

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

[10] J. M. Blackledge, “Digital Image Processing in Radon-Space and the Inversion of Limited Fourier Data”, Optik, vol. 73, no. 2, pp. 74-82, 1986.