Encryption Possessing Statistical Perfect Secrecy and Stealth*

B. John Oommen†

1 Highlights of the Invention

1.1 Patent Information

The details of the patent-protected invention, referred to as DODE*, are as follows:

- **Title of Patent**: A Method For Encryption with Statistical Perfect Secrecy
- **Inventors**: B. John Oommen and Luis G. Rueda[1].
- **Owner of the Patent**: *Oommen Computer Consultants Inc.*, 5942 Third Line Road, North Gower, ON: K0A2T0, Canada.
- **Associated Patents**:
  1. U.S. Patent No. 7,508,935 issued on March 24, 2009.
  2. Canadian Patent No. 2,460,863 issued on April 26, 2011.
- **Significance of the Patent**: This patent solves a compression-based problem that was reported to be unsolved in the standard textbooks. Using this solution, we were able to invent an encryption which provides *Statistical Perfect Secrecy*. We believe that our solution is, to date, the only reported one for this problem.

*The technology has been patent protected as explained in the text. The commercial history of the patent can be discussed separately with interested parties.

†The author can be contacted at: *Chancellor's Professor* and *Fellow: IEEE; Fellow: IAPR*, School of Computer Science, Carleton University, 1125 Colonel By Dr., Ottawa, ON, K1S 5B6, Canada. E-mail: oommen@scs.carleton.ca.

[1]This inventor can be contacted at: *Professor*, School of Computer Science, University of Windsor, 401 Sunset Avenue, Windsor, ON, N9B 3P4, Canada. E-mail: lrueda@uwindsor.ca.

1.2 Overview

The hallmark of a "perfect" cryptosystem is the fundamental property called "Perfect Secrecy" [8]. Informally, this property means that for every input data stream, $\mathcal{X}$, the probability of yielding any given output data stream, $\mathcal{Y}$, is the same, and independent of the input. Consequently, there is no statistical information in the output data stream or Ciphertext, about the identity and distribution of the input data or Plaintext. A system possessing Perfect Secrecy yields an output sequence that is distributed like *random noise*. This means that an eavesdropper who examines the Ciphertext output $\mathcal{Y}$, **cannot** deduce the input $\mathcal{X}$ from analyzing the statistical information in $\mathcal{Y}$.

The problem of attaining Perfect Secrecy was originally formalized by Shannon in 1949 [6]. Shannon [4, 6, 7, 8] showed that if a cryptosystem possesses Perfect Secrecy, then the length of the secret key must be at least as large as the Plaintext. This makes the development of a realistic perfect secrecy cryptosystem impractical, such as demonstrated by the Vernam One-time Pad. Developing a pragmatic encoding system that satisfies this property is an open problem that has been unsolved for many decades.

The invention described here, referred to as DODE*, guarantees *Statistical Perfect Secrecy* - a property closely related to the phenomenon of Perfect Secrecy. A system (including a cryptosystem, a compression system, and in general, an encoding system) is said to possess *Statistical Perfect Secrecy* if all its contiguous output sequences of length k are equally likely, for all values of k, independent of the input data stream, $\mathcal{X}$. Thus, a scheme that removes all statistical properties of the input stream also has the property of Statistical Perfect Secrecy. A system possessing Statistical Perfect Secrecy maximizes the entropy of the output computed on a symbol-wise basis.

It is easy to see that the phenomenon of *Statistical Perfect Secrecy* is related to the concept of Perfect Secrecy. It differs marginally from the former in that it is defined in terms of *all* possible subsequences of $\mathcal{Y}$ of length k (for all k), and not in terms of the entire output sequence $\mathcal{Y}$. Additionally, since the property of Statistical Perfect Secrecy can characterize any system and not just a cryptosystem, there is no requirement of relating the size of the key to the size of the input, as required by Shannon's theorem.

2 The Kernel and Properties of the Invention, DODE*

2.1 The Kernel of the Solution

To obtain a Statistical Perfect Secrecy encryption, we proceed from the first principles of coding theory, Shannon's entropy theory and the theory of stochastic learning, instead of using bit-wise operations (like XORs and the principles of feedback shift registers), elliptic functions, factoring of large numbers, or other NP-hard problems. The goal of the whole exercise is to develop a system generating "white-noise-like" output, where this output or Ciphertext is *statistically independent* of the input or Plaintext.

In the case of DODE*, this is accomplished by solving a problem which we refer to as the *Distribution Optimizing Data Compression (DODC)* problem. This has been reported to be open even in modern-day textbooks [1], and is described briefly in Section 3.

Our solution to the DODC forms the kernel for DODE*.

2.2 The Properties of DODE*

DODE* has the following properties:

1. DODE* is based on the solution of a previously-reported *unsolved* problem.
2. It is a stream cipher using an underlying philosophy that has not been *previously* used in encryption. Indeed, the foundational mathematical tools used in solving the above open problems are those used in the design and analysis of stochastic Learning Automata (LA).
3. DODE* represents the pioneering use of LA in encryption.
4. The scheme encrypts guaranteeing *Statistically* Perfect Secrecy. Thus, the output resembles random noise even for relatively small keys. The Statistical Perfect Secrecy phenomenon is not merely experimentally verified, but also rigorously proven.
5. The cryptosystem passes all the latest FIPS 140-2 tests on the standard benchmark corpuses as explained in Section 2.3.
6. The system also demonstrates bit-wise, key-to-output, and input-output independence on the standard benchmark corpuses.

7. The output $\mathcal{Y}$ maximizes the entropy, $\mathcal{H}(\mathcal{Y})$, *independent* of *any* input data and *any* key.

8. The secret key used by DODE* need not be 128 or 256 bits long. Rather, it can be arbitrarily large. Further, the speed of the encryption decreases *linearly* with the key length, rendering it superior to many modern-day encryption schemes.

9. The scheme guarantees stealth and is thus ideal for steganography.

10. Since the scheme effectively removes all statistical information from the input stream, it does not lend itself to statistics-based cryptanalysis, and can thus be broken only by an exhaustive search of the entire key space.

11. Most reported stream ciphers claim to *experimentally* yield random noise characteristics. However, the strength of DODE* is the *novel underlying philosophy and paradigm* which are distinct from the ones traditionally used, and the rigorous *formal* proofs that it does yield Statistical Perfect Secrecy.

12. The formal details of these proofs can be found in the patent papers for the invention. The patent offices determined that the closest result to our invention was a publication/invention in 1995, but even that solution does not solve the unsolved problem that we have solved. *We emphasize that all the claims that we asked for were evaluated to be novel, inventive, and to possess industrial applicability.*

2.3 Passing the FIPS Industry Stantards

The FIPS Industry Standards are the tests for evaluating the randomness of data, and is the accepted industry standard. To quote from *Wikipedia*:

> " the National Institute of Standards and Technology (NIST) issued the FIPS 140 Publication Series to coordinate the requirements and standards for cryptography modules that include both hardware and software components. Protection of a cryptographic module within a security system is necessary to maintain the confidentiality and integrity of the information protected by the module. This standard specifies the security requirements that will be satisfied by a cryptographic module. The standard provides four increasing, qualitative levels of security intended to cover a wide range of potential applications and environments."

In particular:

1. DODE* passes the **Frequency Test (*monobit* test)**. In this case it passes passes the chi-square *monobit* test with a confidence level higher than 99%.

2. DODE* passes the **Serial Tests (*two-bit tests)***. In this case, it passes the chi-square test with a confidence level higher than 99% for bits which are k-symbols-apart, where $k = 1, \ldots, 5$.

3. DODE* passes all the latest industry-accepted FIPS 140 tests of randomness including the **Poker Test**, the **Runs Tests**, the **Long Runs Test** and the **Autocorrelation Test** [4] (pp. 181-183).

4. The message obtained by "decrypting" using a key that differs from the true key by even a single bit, also has random noise characteristics. This is demonstrated by a suite of key-output tests.

5. Analogous conclusions can be gleaned by examining the results of a suite of input-output independence tests.

These results are found at: `http://www.scs.carleton.ca/~oommen/papers/FIPS-140-2.pdf`.

3 Open Problems

To show how the kernel is developed, we discuss in greater detail the open problem alluded to above.

When data has to be transmitted from one place to another or optimally stored, it is typically compressed and encrypted. The aim is to ensure that a much smaller compressed and secure file can be transmitted or stored, with no loss, to the receiver, who would then decode the encoded file to procure the original file. Viewed from this perspective, there are two fundamental problems involving

(i) *Controlling the output probabilities*, and,

(ii) *Achieving Statistical Perfect Secrecy.*

These two concepts are separately discussed in the next two sub-sections. We have solved the former problem and utilized its solution in DODE* to develop a cryptosystem possessing Statistical Perfect Secrecy and stealth.

3.1 Optimizing the Output Probabilities

Consider a system in which $\mathcal{X} = x[1] \dots x[M]$ is the Plaintext data stream, where each $x[k]$ is drawn from a Plaintext alphabet, $\mathcal{S} = \{s_1 \dots s_m\}$, and $\mathcal{Y} = y[1] \dots y[R]$ is the Ciphertext data stream, where each $y[k] \in \mathcal{A}$ of cardinality r.

There are numerous schemes which have been devised for data compression/encoding. A survey of the field is found in [1, 5, 9].

The problem of obtaining arbitrary encodings of the output symbols has been studied by researchers for at least five decades. Many encoding algorithms (such as those of Huffman, Fano, Shannon, the arithmetic coding and others) have been developed using different statistical and structure models (e.g. dictionary structures, higher-order statistical models and others). They are all intended to compress data, but their major drawback is that they cannot control the probabilities of the output symbols.

This drawback can be explained as follows. Once the data encoding method has been specified, the user loses control of the contents of the encoded file. In other words, in all the compression/encoding techniques known till today, the statistical properties of the output compressed/encoded file cannot be *controlled* by the user - they take on their values as a consequence of the statistical properties of the original file and the data encoding method in question.

A problem that has been open for many decades [1] is that of devising an encoding scheme, which when applied on a data file, compresses the file and *simultaneously makes the file appear to be random noise*. Thus, if the alphabet is binary (as is typically the case) the *input* probability of '0' and '1' could be arbitrary. The problem of the user specifying the *output* probability of '0' and '1' in the encoded file has been open. Indeed, if the user specifies the stringent constraint that the output probabilities of '0' and '1' be arbitrarily close to 0.5, the consequences are very far-reaching, resulting in the erasure of statistical information.

We assume that we are given a code alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$, and that the user specifies the desired output probabilities or frequencies of each $a_j \in \mathcal{A}$ in the encoded file by $\mathcal{F}^* = \{f_1^*, \dots, f_r^*\}$. Such a rendering will be called an "entropy optimal" rendering. On the other hand, the user also simultaneously requires optimal, or even sub-optimal *lossless* data compression. Thus, after compressing a file, we are required to recover **exactly** the same file after the specified decompression process has been invoked. This problem, which we refer to as the "Distribution Optimizing Data Compression" (DODC) problem[2], is known to be

[2]In the more general context of not just compressing the Plaintext, but encoding it, the problem will be

unsolved. Its formal statement is given in Appendix A.

It is interesting to note that the problem is *intrinsically* difficult because the data compression requirements and the output probability requirements can be contradictory. Informally speaking, if the occurrence of '0' is significantly lower than the occurrence of '1' in the original file, designing an encoding method which compresses the file and which simultaneously increases the proportion of '0's to '1's is far from trivial.

The importance of the problem is reflected in the following paragraph as originally stated in [1] (pp. 76-77):

> " It would be nice to have a slick algorithm to solve Problem 1, especially in the case $r = 2$, when the output will not vary with different definitions of $d(\mathcal{F}, \mathcal{F}^*)$. Also, the case $r = 2$ is distinguished by the fact that binary channels are in widespread use in the real world.
>
> We have no such algorithm! Perhaps someone reading this will supply one some day..."

3.2 Statistical Perfect Secrecy

As mentioned in Section 1.2, the problem of erasing the statistical distribution from the input data stream in the encoded output data stream, has fundamental significance in cryptographic applications. It is well known that any good cryptosystem should generate an output that has random characteristics [4, 6, 7, 8].

A fundamental goal in cryptography is to attain Perfect Secrecy [8]. Indeed, as mentioned earlier, it is well known that it is quite impractical to develop a cryptosystem possessing Perfect Secrecy because such a system must have a secret key whose length is at least as large as the size of the Plaintext.

The phenomenon of Statistical Perfect Secrecy eliminates the constraint between the key length and the length of the Plaintext. This is achieved by defining the corresponding probabilities in terms of the subsequences of arbitrary lengths, as opposed to the entire output Ciphertext. Thus, a system (including a cryptosystem, a compression system, and in general, an encoding system) is said to possess *Statistical Perfect Secrecy* if all its contiguous output sequences of length k are equally likely, for all values of k, independent of $\mathcal{X}$.

More formally, a system is said to possess Statistical Perfect Secrecy if for every input $\mathcal{X}$ there exists some integer $j_0 \geq 0$ and an arbitrarily small positive real number δ_0 such that for all $j > j_0$,$\Pr[y_{j+1}...y_{j+k}|\mathcal{X}] = \frac{1}{r^k} \pm \delta_0$ for all k, $0 < k < R - j_0$.

referred to as the Distribution Optimizing Data Encoding (or "DODE") problem.

If the encoding system can guarantee this property, we can assert that for all practical purposes and for all finite-lengthed subsequences, statistically speaking, the system behaves as if it, indeed, possessed the stronger property of *Perfect Secrecy*.

3.3 Solving these Open Problems

Our current solution to both these open problem uses stochastic learning methodologies to achieve this encoding. The solution is both optimal and lossless. Additionally, the scheme is adaptive, and so the *input* probabilities need not be estimated. Finally, the output probabilities are controllable so that the output quickly converges to the specified distribution of '0's and '1's. All of these properties have been theoretically proven and experimentally verified to demonstrate adherence to the highest industry standards.

This algorithm achieves amazing results for *any* file type, and with *any* probability distribution.

4 Solution Proposed for these Open Problems

4.1 Highlights of the Solution

We have solved the open DODC problem stated as Problem 1 in the Appendix for the case when the input alphabet is any r-ary alphabet and the output alphabet is binary. This solution can easily be extended to the multi-symbol output alphabet case.

4.2 Solution for General Encoding Systems

Apart from solving the reported open DODC problem (as stated as Problem 1 in the Appendix), we have also solved the more general problem when the output probabilities are requested by the user, but where s/he does not require the output to be *compressed*. Indeed, in the most general setting, the user may require that while the output is entropy optimal, the output size is simultaneously maintained the same, or even increased. This represents the general Distribution Optimizing Data Encoding (DODE) problem.

By extending our solution to the basic DODC problem, we have also solved the DODE problem. The latter solution involves a new data structure referred to as the Oommen-Rueda Structure. Adaptively blending the exponential number of Oommen-Rueda Structures leads to specific structure-based restructuring rules, which, in turn, lead to formal techniques for

entropy-optimal encoding and decoding processes. Indeed, using these rules, we have been able to solve the DODE problem for the most general case when the input alphabet is of cardinality m, and the output alphabet is of cardinality r.

4.3 Uniqueness of the Solution

We have used the DODC solution (or in the most general setting, the DODE solution) as the kernel to our encryption. To get an overall perspective of the uniqueness and novelty of our contribution, we refer to the following statements from [4] (pp. 191-192):

(*i*) There are relatively few fully specified stream ciphers in the literature. This is because most stream ciphers used, in practice, tend to be proprietary and confidential.

(*ii*) Feedback Shift Registers (FSR), and, in particular, Linear Feedback Shift Registers (LFSR) are the basic building blocks in most stream ciphers.

(*iii*) LFSRs can be combined using either nonlinear combining functions, nonlinear filtering functions, or using the output of one or more LFSRs to control the clock of one or more LFSRs.

(*iv*) Clock-controlled stream ciphers are also reported in the literature, examples of which are the alternate step generator and the shrinking generator.

In contrast to these, our invention, DODE*:

(*i*) Does not utilize XORs or FSRs in *any* form as a basic building block.

(*ii*) Is not clock-controlled in any manner.

(*iii*) Is based on the solution to the open DODC problem mentioned above, which till today has not been used in encryption. *This philosophy lends itself as the new paradigm based on which we have designed our novel stream ciphers.*

(*iv*) Although DODE* and its cryptographic enhancements are not FSR-based, we believe that they can be easily implemented in hardware.

(*v*) Most reported stream ciphers claim to *experimentally* yield random noise characteristics. However, the strength of DODE* is the *novel underlying philosophy and paradigm* which are distinct from the ones traditionally used, and the rigorous *formal* proofs that the schemes do yield Statistical Perfect Secrecy.

(*vi*) The formal details of these proofs can be found in the patent papers for the invention. As mentioned in Section 2.2, *all the claims we asked for were evaluated to be novel, inventive, and to possess industrial applicability.*

This affirms the uniqueness and the novelty of our invention!

5 Cryptanalysis of the Invention

To visit the issue of cryptanalyzing DODE*, we briefly cite the following points from [3]:

(a) The most typical use of a stream cipher for encryption is to generate a keystream in a way that depends on the secret key, and then to combine this (typically using bit-wise XORs) with the message being encrypted.

(b) It is imperative that the keystream "looks" random, and that this is verified by passing a battery of statistical tests.

(c) A good stream cipher utilizes a keystream which has a very large period. Thus, if the period of the keystream is too short, the adversary can apply discovered parts of the keystream to help in the decryption of other parts of the Ciphertext.

(d) A more involved set of structural weaknesses might offer the opportunity of finding alternative ways to generate part, or even the whole of the keystream. This gives rise to the measure of security known as the *linear complexity* of a sequence, where the linear complexity is the size of the LFSR that needs to be used to replicate the sequence. The authors of [3] state that a necessary condition for the security of a stream cipher is that the sequences it produces have a high linear complexity.

(e) Other attacks attempt to recover part of the secret key that was used. Apart from the most obvious attack of searching for the key by brute force, a powerful class of attacks can be described by the term divide and conquer.

Bearing these in mind, it should be mentioned that DODE* is not vulnerable to the typical cryptanalytic methods used for stream ciphers for the following reasons:

(a) As opposed to the currently available stream ciphers, DODE* obtains the Ciphertext by utilizing the key in effecting the structure-based restructuring rules, and thereafter operating on the Plaintext to generate the Ciphertext.

(b) It should be emphasized that this is achieved in a single process which cannot be decomposed. DODE* is, indeed, a *stream cipher* because it processes the Plaintext in a symbol-wise manner.

(c) With regard to the rigorous statistical tests that measure the correlation of bits in the Ciphertext, we see from Section 2.3 that the output of DODE* effectively passes all the most-recent FIPS 140 tests.

(d) Divide-and-conquer methods of cryptanalysis will not be effective for DODE* since its output passes rigid input-output and key-output statistical independence tests.

(e) With regard to the *period* of DODE*, we observe that although we have not yet determined the period of Ciphertexts generated by DODE*, it can be easily shown that this period is *strictly greater* than that of *any* of the cryptographically-secure pseudo-random number generators reported.

(f) The *linear complexity* of the Ciphertext generated by DODE* has not been computed. Furthermore, since DODE* does not use FSRs in any form, but utilizes techniques that have not been previously used in designing stream ciphers, we believe that this complexity is not easily computable, either. However, since the period of the output of DODE* is very large, and since the output passes the rigid input-output and key-output statistical independence tests, we contend that the linear complexity of the Ciphertext generated by DODE* is also correspondingly high.

Thus, DODE* is not vulnerable to the traditional techniques for attacking stream ciphers.

On the other hand, since other well-known cryptanalytic techniques, such as *linear cryptanalysis* and *differential cryptanalysis* are not typically useful in breaking stream ciphers (but rather, block ciphers), we argue that they will not assist in cryptanalyzing DODE* either. Indeed, the axioms that warrant these methods imperatively utilize the statistical information resident in the Ciphertext.

DODE* cannot be broken using linear cryptanalysis, because this technique is based on the non-uniformity of the output distribution. This non-uniformity is the characteristic that is erased using the DODC kernel and its enhancement, DODE*.

Similarly, differential cryptanalysis is also based on the non-uniformity of the distribution of the data streams resulting from performing XOR (or similar) operations. Since (a) the output of DODE* possesses the Statistical Perfect Secrecy property, (b) the output of DODE* demonstrates independence for higher-order Markovian models, (c) the blending achieved by the stochastic learning is adaptive, and (d) the output empirically demonstrates an independence of the input, the resulting output obtained from performing XOR (or similar) operations between output sequences will *also* demonstrate random noise characteristics.

Thus, DODE* will not be vulnerable to differential cryptanalysis.

6 Applications of DODE*-based Solutions

Humanity has been sending secret messages for centuries. Our generation is not unique in this. One way to develop "harder-to-break" encryptions is to use increasingly larger keys. However, the real art and science of advancing the field is to know how to design encryptions with novel technologies that have not been previously used. This is the case with DODE*, and thus:

1. It invokes a completely new theory and so the cryptanalysis techniques will have to be new.

2. Not only is the cryptanalysis unknown, but more importantly, *even the methods to be used to achieve the cryptanalysis are unknown.* Since, it is a relatively new patent, as far as we know, people have not started attempting to break it.

In this light, we now list the primary applications of DODE*:

1. **Secure Data Storage**: Consider problem of storing data on the "*Cloud*". In such cases, one is not aware of where (i.e., the servers) the data resides. One is often not even aware of *who* controls the data. Finally, the user is not even aware of how the firewalls of the servers are maintained. If now the firewalls of the servers in the "*Cloud*" are compromised, the Intruder would never be able to access the data itself if it were encoded using DODE*. Indeed, the Intruder would only see a file populated with random noise.

- We have built two (JAVA and C) applications that achieve this encoding on a typical modern day desktop equipped with a 3.6 GHz processor.
- The applications can encode 1GigByte of data in about 100 seconds.
- The potential of using this desktop application for secure storage in the "*Cloud*" is virtually unbounded.

2. **Secure Messaging**: Messaging using mobile Apps are becoming increasingly popular (for example, using WhatsApp, Telegram etc.). DODE* can be used to allow secure mobile messaging and communication. Again, in the absence of the secret key, an Eavesdropper would never only see random noise.

   - We have built a prototype mobile App for Android devices, which has all the standard messaging functionalities of a typical mobile App.
   - The prototype is able to send messages, include emoticons, and also attach pictures etc..

3. The other major applications are those that arise from DODE*'s cryptographic capabilities, and include:

   - Traffic on the internet.
   - E-commerce/e-banking.
   - E-mail/ftp.
   - Video conferencing.
   - Audio/Video streaming.
   - secure communications, including Secure IP, Secure VOIP, Secure Satellite, secure mobile and wireless communications.
   - Watermarking, in which the watermark is generated by invoking a DODE*-based encryption.
   - Steganography, which is the ancient art of hiding information – an art that dates back from around 440 B.C [2]. One of the most common steganographic techniques consists of hiding information in images. To experimentally demonstrate the power of DODE*, it has been incorporated into a steganographic application by modifying 256-gray scale images with DODE*'s random-noise output. By

virtue of the DODE*'s Statistical Perfect Secrecy property, its use in steganography is unique. An example of this is shown in Figure 1. We first display the original Lena image. We then display the image resulting from embedding the output obtained from encrypting the file *fields.c* of the Canterbury corpus. Apart from the two images being visually similar, their respective histograms pass the similarity test with a very high level of confidence.

Figure 1: The original carrier image and the resulting image after applying steganographic techniques to the output with an encoding of the file *fields.c* from the Canterbury corpus. The steganographic method used is the fairly straightforward Least Significant Bit approach.

7 Conclusions

In this report we have presented a novel method of encryption whose kernel is based on a solution to a problem which has been reported open, namely the Distribution Optimizing Data Compression (DODC) problem. The invention, referred to as DODE*, guarantees *Statistical Perfect Secrecy* - a property which is closely related to the phenomenon of Perfect Secrecy. The Statistical Perfect Secrecy phenomenon is not merely experimentally verified for the most recent FIPS 140 tests on the standard benchmark corpuses. It has also been formally and rigorously proven. The resulting cryptosystem also demonstrates bit-wise, key-to-output, and input-output independence on the standard benchmark corpuses.

The speed of the encryption decreases *linearly* with the key length, rendering it superior to many modern-day encryption schemes. One of the embodiments of the invention achieves on-line data compression, and thus permits communication with an increased bandwidth.

Finally, since the scheme effectively removes all statistical information from the input stream, it does not lend itself to statistics-based cryptanalysis, and can thus be broken only by an exhaustive search of the entire key space.

The report also includes a list of the possible applications of DODE*.

References

[1] D. Hankerson, G. Harris, and P. Johnson Jr. *Introduction to Information Theory and Data Compression*. CRC Press, 1998.

[2] S. Katzenbeisser and F. Peticolas. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

[3] RSA Laboratories. *RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1*. RSA Security Inc., 2000.

[4] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[5] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, 2nd. edition, 2000.

[6] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communications*. University of Illinois Press, 1949.

[7] W. Stallings. *Cryptography & Network Security: Principles & Practice*. Prentice Hall, 1998.

[8] D. R. Stinson. *Cyptography : Theory and Practice*. CRC Press, 1995.

[9] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 2nd. edition, 1999.

Appendix A: Formal Statement of the Problems

As stated in [1], the Distribution Optimizing Data Compression (DODC) problem can be more formally written as follows:

Problem 1. Consider the source alphabet, $\mathcal{S} = \{s_1, \ldots, s_m\}$, with probabilities of occurrence, $\mathcal{P} = [p_1, \ldots, p_m]$, an input sequence, $\mathcal{X} = x[1] \ldots x[M]$, the code alphabet, $\mathcal{A} = \{a_1, \ldots, a_r\}$, and the optimal frequencies, $\mathcal{F}^* = \{f_1^*, \ldots, f_r^*\}$, of each a_j, $j = 1, \ldots, r$, required in the output sequence. The output is to be a sequence, $\mathcal{Y} = y[1] \ldots y[R]$, whose probabilities are $\mathcal{F} = [f_1, \ldots, f_r]$, generated by the encoding scheme $s_i \rightarrow w_i \in \mathcal{A}^+$ such that:

(*i*) The scheme must generate a *prefix* code, required for lossless compression.

(*ii*) The average code word length of the encoding scheme, $\overline{\ell} = \sum_{i=1}^{m} p_i \ell_i$, must be *minimal* among all the prefix codes, where ℓ_i is the length of w_i.

(*iii*) The distance, $d(\mathcal{F}, \mathcal{F}^*) = \sum_{j=1}^{r} |f_j - f_j^*|^\alpha, \alpha \geq 1$ must be *minimized*.

Typically, the distance is measured using the *mean square error*, by setting $\alpha = 2$.

The problem of achieving *Perfect Secrecy* is stated below.

Problem 2. Let $\mathcal{X} = x[1] \ldots x[M]$ be the Plaintext, where each $x[k]$ is drawn from a source alphabet, $\mathcal{S} = \{s_1 \ldots s_m\}$. Let $\mathcal{Y} = y[1] \ldots y[R]$ be the Ciphertext, where each $y[k] \in \mathcal{A}$, the Ciphertext alphabet.

A cryptosystem has *Perfect Secrecy* if $\Pr[\mathcal{X}|\mathcal{Y}] = \Pr[\mathcal{X}]$.

By Bayes theorem, this is equivalent to requiring that $\Pr[\mathcal{Y}|\mathcal{X}] = \Pr[\mathcal{Y}]$.