International Journal of Research Studies in Computing
2012 April, Volume 1 Number 1, 35-42

An application of data encryption technique using random number generator

Verma, Sharad Kumar
Mewar University, Rajasthan, India (sharadverm@gmail.com)
Ojha, D. B.
Mewar Institute of Technology, Ghaziabad, UP, India (ojhabrat@gmail.com)

ISSN: 2243-772X
Online ISSN: 2243-7797

OPEN ACCESS

Received: 19 January 2012 **Revised**: 5 February 2012 **Accepted**: 7 February 2012
Available Online: 11 February 2012 **DOI:** 10.5861/ijrsc.2012.v1i1.72

Abstract

Coding theory is one of the most important and direct applications of information theory. Using a statistical description for data, information theory quantifies the number of bits needed to describe the data, which is the information entropy of the source. Information theoretic concepts apply to cryptography and cryptanalysis. Cryptography is the study of sending and receiving secret messages. With the widespread use of information technologies and the rise of digital computer networks in many areas of the world, securing the exchange of information has become a crucial task. In the present paper an innovative technique for data encryption is proposed based on the random sequence generation. The new algorithm provides data encryption at two levels and hence security against crypto analysis is achieved at relatively low computational overhead.

Keywords: coding theory; cryptography; cryptanalysis; encryption; entropy

© The Authors / Attribution-NonCommercial-NoDerivs CC BY-NC-ND

An application of data encryption technique using random number generator

1. Introduction

Cryptography or *cryptology*; from Greek meaning "hidden, secret"; and "writing", or "study" respectively; is the practice and study of techniques for secure communication in the presence of third parties called adversaries (Liddell & Scott, 1984). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries (Bellare & Rogaway, 2005) and which are related to various aspects in information security such as data confidentiality, data integrity, and authentication (Menezes, van Oorschot, & Vanstone, 1997). Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

1.1 Cryptographic goals

Generally, a good cryptography scheme must satisfy a combination of four different goals (Cole, Fossen, Northcutt, & Pomeranz, 2003).

1. *Authentication*: Allowing the recipient of information to determine its origin, that is, to confirm the sender's identity. This can be done through something you know or you have. Typically provided by digital signature.

2. *Non-repudiation*: Ensuring that a party to a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated. Typically provided by digital signature.

3. *Data integrity*: A condition in which data has not been altered or destroyed in an unauthorized manner. Typically provided by digital signature.

4. *Confidentiality*: Keeping the data involved in an electronic transaction private. Typically provided by encryption

There are two main types of cryptography. Those are public-key and symmetric-key. Public-key is a form of cryptography in which two digital keys are generated, one is private, which must not be known to another user, and one is public, which may be made available in public. These keys are used for either encrypting or signing messages. The public-key is used to encrypt a message and the private-key is used to decrypt the message. However, in another scenario, the private-key is used to sign a message and the public-key is used to verify the signature. The two keys are related by a hard one-way (irreversible) function, so it is computationally infeasible to determine the private key from the public key. Since the security of the private key is critical to the security of the cryptosystem, it is very important to keep the private key secret. This public-key system has the problem of being slow.

On the other hand, the system has powerful key management and, even more importantly, public-key cryptography has the ability to implement digital signatures in an efficient way. However, symmetric-key is a form of cryptography in which two parties that want to communicate can share a common and secret key. Each party must trust the other not to tell the common key to anyone else. This system has the advantage of encrypting large amount of data efficiently. However, the problem arises when it comes to key management over large number of users (Certicom Corp, 2004).

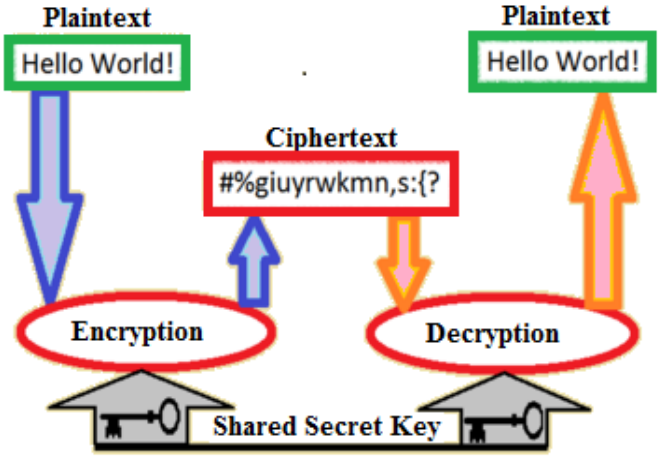


Figure 1. Symmetric-key cryptography, where the same key is used both for encryption and decryption (Chandra Sekhar, Sudha, & Prasad Reddy, 2007)

Coding theory is one of the most important and direct applications of information theory. It can be subdivided into source coding theory and channel coding theory. Using a statistical description for data, information theory quantifies the number of bits needed to describe the data, which is the information entropy of the source.

1. *Data compression* (source coding): There are two formulations for the compression problem:

   A. Lossless data compression: the data must be reconstructed exactly;

   B. Lossy data compression: allocates bits needed to reconstruct the data, within a specified fidelity level measured by a distortion function. This subset of Information theory is called rate–distortion theory.

2. *Error-correcting codes* (channel coding): While data compression removes as much redundancy as possible, an error correcting code adds just the right kind of redundancy (i.e., error correction) needed to transmit the data efficiently and faithfully across a noisy channel.

This division of coding theory into compression and transmission is justified by the information transmission theorems, or source–channel separation theorems that justify the use of bits as the universal currency for information in many contexts. However, these theorems only hold in the situation where one transmitting user wishes to communicate to one receiving user. In scenarios with more than one transmitter (the multiple-access channel), more than one receiver (the broadcast channel) or intermediary "helpers" (the relay channel), or more general networks, compression followed by transmission may no longer be optimal. Network information theory refers to these multi-agent communication models.

Figure 2. A picture showing scratches on the readable surface of a CD-R. Music and data CDs are coded using error correcting codes and thus can still be read even if they have minor scratches using error detection and correction.

1.2 Source theory

Any process that generates successive messages can be considered a **source** of information. A memory-less source is one in which each message is an independent identically-distributed random variable, whereas the properties of ergodicity and stationarity impose more general constraints. All such sources are stochastic. These terms are well studied in their own right outside information theory.

1.3 Rate

Information **rate** is the average entropy per symbol. For memory-less sources, this is merely the entropy of each symbol, while, in the case of a stationary stochastic process, it is

$$r = \lim_{n \to \infty} H(X_n | X_{n-1}, X_{n-2}, X_{n-3}, \ldots);$$

That is, the conditional entropy of a symbol given all the previous symbols generated. For the more general case of a process that is not necessarily stationary, the *average rate* is

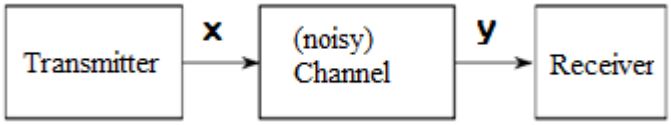
$$r = \lim_{n \to \infty} \frac{1}{n} \mathrm{H}(X_1, X_2, \ldots, X_n);$$

That is, the limit of the joint entropy per symbol. For stationary sources, these two expressions give the same result.

It is common in information theory to speak of the "rate" or "entropy" of a language. This is appropriate, for example, when the source of information is English prose. The rate of a source of information is related to its redundancy and how well it can be compressed, the subject of ***source coding***.

1.4 Channel capacity

Communications over a channel - such as an ***Ethernet cable*** - is the primary motivation of information theory. As anyone who's ever used a telephone (mobile or landline) knows, however, such channels often fail to produce exact reconstruction of a signal; noise, periods of silence, and other forms of signal corruption often degrade quality. How much information can one hope to communicate over a noisy (or otherwise imperfect) channel? Consider the communications process over a discrete channel. A simple model of the process is shown below:



Here X represents the space of messages transmitted, and Y the space of messages received during a unit time over our channel. Let $p(y \mid x)$ be the conditional probability distribution function of Y given X. We will consider $p(y \mid x)$ to be an inherent fixed property of our communications channel (representing the nature of the **noise** of our channel). Then the joint distribution of X and Y is completely determined by our channel and by our choice of $f(x)$, the marginal distribution of messages we choose to send over the channel. Under these constraints, we would like to maximize the rate of information, or the **signal**, we can communicate over the channel. The appropriate measure for this is the mutual information, and this maximum mutual information is called the **channel capacity** and is given by:

$$C = \max_{f} \mathrm{I}(\mathrm{X}; \mathrm{Y}).$$

This capacity has the following property related to communicating at information rate R (where R is usually bits per symbol). For any information rate $R < C$ and coding error $\varepsilon > 0$, for large enough N, there exists a code of length N and rate $\geq$ R and a decoding algorithm, such that the maximal probability of block error is $\leq \varepsilon$; that is, it is always possible to transmit with arbitrarily small block error. In addition, for any rate $R > C$, it is impossible to transmit with arbitrarily small block error.

Channel coding is concerned with finding such nearly optimal codes that can be used to transmit data over a noisy channel with a small coding error at a rate near the channel capacity.

2. Data encryption using random number

A cipher is an algorithm for performing encryption (and the reverse, decryption) - a series of well-defined steps that can be followed as a procedure. Classical ciphers are based around the notions of character substitution and transposition. Messages are sequences of characters taken from some plaintext alphabet (e.g. the letters A to Z) and are encrypted to form sequences of characters from some cipher text alphabet. The plaintext and cipher text alphabets may be the same. Substitution ciphers replace plaintext characters with cipher text characters. For example, if the letters of the alphabet A . . Z are indexed by 0 . . . 25, then a Caesar cipher might replace a letter with index k by the letter with index (k + 3) mod 26. Thus, the word "JAZZ" would become "MDCC". Transposition ciphers work by shuffling the plaintext in certain ways. Thus, reversing the order of letters in successive blocks of four would encrypt "CRYPTOGRAPHY" as "PYRCRGOTYHPA".

Modern crypto-systems have now supplanted the classical ciphers but cryptanalysis of classical ciphers is the most popular cryptological application for meta-heuristic search research. The reasons are probably mixed. The basic concepts of substitution and transposition are still widely used today (though typically using blocks of bits rather than characters) and so these ciphers form simple but plausible test beds for exploratory research. Problems of varying difficulty can easily be created (e.g. by altering the key size). One cannot know how correct a decrypted text is without knowing the plaintext. Instead, the degree to which decrypted text has the distributional properties of natural language is taken as a surrogate measure of correctness of the decryption key. In English text the letter "E" will usually occur more than any other. Similarly, the pair (bigram) "TH" will occur frequently, as will the triple (trigram) "THE".

In contrast, the occurrence of the pair "AE" is less common and the occurrence of "ZQT" is either a rare occurrence of an acronym or else indicates a terrible inability to spell. The frequencies with which these various N-grams appear in plaintext are used as the basis for determining the correctness of the key which produced that plaintext. The more the frequencies resemble expected frequencies, the closer the underlying decryption key is assumed to be to the actual key. With probabilistic encryption algorithms, a crypto analyst can no longer encrypt random plain texts looking for correct cipher text. Since multiple cipher texts will be developed for one plain text, even if he decrypts the message to plain text, he does not know how far he had guessed the message correctly. Also the cipher text will always be larger than plain text.

The new encryption algorithm is based on the concept of Poly alphabet cipher, which is an improvement over mono alphabetic technique. In this technique the character in the plain text is replaces using a random sequence generator. Random number generator using quadruple vector: For the generation of the random numbers a quadruple vector is used. The quadruple vector T is generated for 44 values i.e. for 0-255 ASCII values.

T = [0 0 0 0 0 0 0 0 1 1 ..……3
0 0 0 0 1 1 1 1 2 2 ……..3
0 1 2 3 0 1 2 3 0 1 ……..3]

The recurrence matrix [1] [2]

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

is used to generate the random sequence for the 0-255 ASCII characters by multiplying r = [A] * [T] and considering the values to mod 4.

The random sequence generated using the formula [40 41 42]*r is as follows:

Random = [0 16 32 48 5 21 37 53 10 26 42 58 15 31 47 63 4
20 36 52 9 25 41 57 14 30 46 62 3 19 35 51 8 4 40 56
13 29 45 61 2 18 34 50 7 23 39 55 12 28 44 60 1 17 33
49 6 22 38 54 11 27 43 59 0 16 32 48 5 21 37 53 10 26
42 58 15 31 47 63 4 20 36 52 9 25 41 57 14 30 46 62 3
19 35 51 8 24 40 56 13 29 45 61 2 18 34 50 7 23 39 55
12 28 44 60 1 17 33 49 6 22 38 54 11 27 43 59 0 16 32
48 5 21 37 53 10 26 42 58 15 31 47 63 4 20 36 52 9 25
41 57 14 30 46 62 3 19 35 51 8 24 40 56 13 29 45 61 2 18
34 50 7 23 39 55 12 28 44 60 1 17 33 49 6 22 38 54 11 27
43 59 0 16 32 48 5 21 37 53 10 26 42 58 15 31 47 63 4
20 36 52 9 5 41 57 14 30 46 62 3 19 35 51 8 24 40 56
13 29 45 61 2 18 34 50 7 23 39 55 12 28 44 60 1 17 33 49
6 22 38 54 11 27 43 59]

3. Encryption and decryption process:

To avoid the result to be guessed by combination and permutation, we can offset the result by some simple rules, as shown in the following (Chandra Sekhar, Sudha, & Prasad Reddy, 2007):

Case 1. Offset by constant value:

HOW ARE U+ n (e.g. n=10) = RYa*K[O*CY-

Case 2. Offset by a polynomial function:

HOW ARE U + [38 37 36 35 34 33 32 31 30]

In the above two equations, we offset the original result by a different methodology and apply 255 sets (1 to 255 except for 0 for convenience in calculating) of ASCII codes to position the new result. If the results of these offset operations are over 255, it will be adjusted. For instance, the ASCII number of the E element is 72, 72 added to 3 8 makes 6633, the value is adjusted to 255, i.e. 6633 -26*255 = 3, the new output therefore becomes ASCII character ‘‘ETX’’. These methods are so easy to finish by a program and their decoding operations are also easy to accomplish by inverting the above operation, i.e. 3 + 26* 255 - 38 =72. In general the equation would be $1 \leq$ ASCII number of encoded character + x.255 – offset rule $\leq$ 255.

3.1 Algorithm

1. A recurrence matrix used is as a key. Let it be A.
2. Generate a "quadruple vector" T for 44 values, i.e, from 0 to 255.
3. Multiply r= A *T;
4. Consider the values to mod 4.
5. A sequence is generated using the formula [40 41 42]*r.
6. This sequence is used as a key
7. Convert the plain text to equivalent ASCII value.
8. Add the key to the individual numerical values of the message
9. New offset the values using the offset rules
10. This would be the cipher text generated
11. For Decryption the key is subtracted from the cipher text and use the offset rule to get the original message

3.2 Example:

3.2.1 Encryption

1. Plain text: HOW ARE U
2. The equivalent ASCII characters are [72 79 87 32 65 82 69 32 89 79 85]
3. From the random sequence the key is chosen as [0 16 32 48 5 21 37 53 10 26 42]
4. Adding the key to the equivalent ASCII string of the plain text we get Ci = [72 95 119 80 70 103 106 85 99 105 127]
5. Using the offset rule2
6. C'i1 = [59121 19778 6680 2267 799 346 187 112 108 108 128]
7. Adjusting to 255 we get C'i1 = [216 143 50 227 34 91 187 112 108 108 128]
8. Hence the cipher text would be ‡ Ă 2 π " [╗ p 11 Ç

3.2.2 Decryption

1. C'i1 = [216 143 50 227 34 91 187 112 108 108 128]
2. By using the offset rule we get Ci = [72 95 119 80 70 103 106 85 99 105 127]
3. Subtracting the key from the cipher text we get [72 79 87 32 65 82 69 32 89 79 85]
4. Which is the chosen plain text: HOW ARE U

4. Conclusions

In the new algorithm a quadruple vector is considered. A mod function is used on the product of matrix key and the quadruple vector. Thus the computational overhead is very low. It is almost impossible to extract the original information in the proposed method even if the algorithm is known. In block cipher algorithms, the plain text is converted into cipher text after a number of rounds, which makes the computational more complex.

5. References:

Bellare, M. & Rogaway, P. (2005). *Introduction to modern cryptography*. Retrieved January 5, 2012, from http://cseweb.ucsd.edu/~mihir/cse207/w-intro.pdf

Certicom Corp. (2004). *The elliptic curve cryptosystem for smart cards*. Retrieved January 5, 2012, from http://www.sans.org/reading_room/whitepapers/vpns/elliptic-curve-cryptography-smart-cards_1378

Chandra Sekhar, A., Sudha, K. R., & Prasad Reddy, PVGD. (2007). Data encryption scheme using random

number generation. *2007 IEEE International Conference on Granular Computing* (pp. 569-579), San Jose, CA: USA.

Cole, E., Fossen, J., Northcutt, S., & Pomeranz, H. (2003). *SANS security essentials with CISSP CBK Version 2.1*.USA: SANS Press.

Liddell, H. G., & Scott, R. (1984). *Greek-English lexicon*. Oxford University Press.

Menezes, A. J., van Oorschot, P. C. & Vanstone, S. A. (1997). *Handbook of applied cryptography* (vol. 6). CRC Press.

Rivest, R. L. (1990). Cryptology. In J. Van Leeuwen, *Handbook of theoretical computer science* (vol. 1, pp. 717-715). Amsterdam: Elsivier.