Information Sciences 270 (2014) 288–297

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Gray code permutation algorithm for high-dimensional data encryption

Massimiliano Zanin [a,*], Alexander N. Pisarchik [b,c]

[a] INNAXIS Foundation & Research Institute, 28006 Madrid, Spain
[b] Centro de Investigaciones en Optica, Loma del Bosque 115, Lomas del Campestre, 37150 Leon, Guanajuato, Mexico
[c] Centre for Biomedical Technology, Technical University of Madrid, Campus Montegancedo, 28223 Pozuelo de Alarcón, Madrid, Spain

ARTICLE INFO

Article history:
Received 21 July 2009
Received in revised form 5 January 2011
Accepted 15 February 2014
Available online 4 March 2014

Keywords:
Image encryption
Chaotic cryptosystem
Permutation box
Gray code

ABSTRACT

We present a novel permutation algorithm for fast encryption of a large amount of data, such as 3D images and real-time videos. The proposed *P-Box* algorithm takes advantage of Gray code properties and allows fast encryption with high information diffusion. The algorithm is optimized for integer q-bit operations ($q = 8, 16, 32, \ldots$), allowing a direct implementation in almost any hardware platform, while avoiding rounding errors of floating-point operations. By combining the *P-Box* with chaotic *S-Box* based on the logistic map, we design a complete, highly secure and fast cryptosystem.

1. Introduction

In recent years, the volume of information that needs to be transmitted through any communication media has gone up very rapidly; and while security still remains a serious problem, speed is becoming another important issue. There is an evident relationship between the encryption/decryption time (EDT) and the quantity of information to be transmitted; the larger the data size, the longer the EDT. New multimedia have to show off the limitations of standard encryption schemes [22,5], since they require very large computational time, restricting their implementation for real-time signal transmission. To increase security, some algorithms use several encryption loops (see, e.g., [13]) calling for larger EDT. The challenge of modern cryptography is to optimize the balance between security and encryption speed, while dealing with a bigger and bigger volume of information.

It is well known that a strong cryptosystem must include two main steps [15]: *P-Box* or permutation step, where the information position is changed in the data sequence, and *S-Box* or substitution step, where every single piece of information (symbol or group of symbols, e.g. each byte) is substituted by another symbol. These two steps reflect two basic properties of a good cryptosystem, confusion and diffusion [23]. When one deals with a large amount of information (for example, real-time videos or 3D images), several fast algorithms are available for implementing the *S-Box*; they usually encode groups of bits in a sequential way, for instance, any *stream ciphers*, such as A5/1 or RC4 [15,24]. It is with the *P-Box* processes that serious speed problems arise, since these algorithms need to shuffle the whole message many times, following complex rules and therefore rising the computational cost.

Recently, many new secure algorithms which use nontraditional techniques have been developed (see, for instance, [16,9,17,26,4,3]). Some of them are based on discrete chaotic systems [16,9,17], because of the ergodic property and their

* Corresponding author. Tel.: +34 606 993761.
E-mail address: massimiliano.zanin@ctb.upm.es (M. Zanin).

http://dx.doi.org/10.1016/j.ins.2014.02.131
0020-0255/

high sensitivity to initial conditions and parameters they provide both *P-Box* and *S-Box* processes with good efficiency and high security. Among possible approaches for high-dimensional data encryption with chaotic maps, we distinguish the most secure algorithms based on the implementation of map lattices' layers [18,19] and on the use of spatial chaotic maps [25]. As in many other cryptographic steps (or algorithms), theoretically the best result would be a completely random output. Indeed, the only algorithm which has been mathematically proven as secure is the "one time pad" encryption scheme. If permutations (or substitution) are performed on a random number sequence, there is no way to recover the original message, because of the lack of structure in the encrypted random data. Of course, this approach is usually not feasible in practice, since it requires a key (a random sequence) as large as the message to be encrypted. Recently, Sun et al. [25] designed a high secure permutation algorithm based on the logistic map; however the authors were unmindful of the high computational cost of their cipher, that makes their cryptosystem unpractical for real-time applications. Similar problems arise with traditional cryptographic techniques, such as DES and IDEA [22,5], while handling new multimedia formats; they are no longer suitable for practical image encryption, especially in real-time communication scenarios [1].

The purpose of this work is to design a rapid *P-Box* algorithm which would allow a fast permutation of a very large amount of data inside a multi-dimensional memory structure. It is clear that a cryptosystem based on permutation only, cannot guarantee much security because of their vulnerability to plaintext attacks; to ensure high security, a complete cryptosystem needs a secure *S-Box* [12]. Even so, a designed cryptosystem using both a *P-Box* and an *S-Box* may not be good enough to deter attacks; for example, one such system has been cryptanalyzed by Rhouma and Belghith [20]. If we are willing to pay the security tag in favor of speed as far as *P-Box* is concerned, the security of the complete cryptosystem has to rely completely on the security of the *S-Box* selection, one good possibility is an *S-Box* based on chaotic systems, such as the one we recently proposed [19]. The main function of the designed *P-Box* is the fast permutation of very extensive information data. To approach this problem, we take advantages of Gray numbers, as an alternative for existing chaotic algorithms proven inefficient when in need to manipulate a large amount of data in real time. We then demonstrate the capacity of such *P-Box* for high-dimensional data encryption on a real 3D image. We also test the security of a complete cryptosystem build with the designed *P-Box* and a chaotic map based *S-Box*.

The rest of the paper is organized as follows. In Section 2 we demonstrate the inefficiency of a *P-Box* based on chaotic maps for encryption of a large amount of data. In Section 3 we construct a new fast permutation algorithm based on the Gray code, in Section 4 we demonstrate its computational benefits when encrypting real 3D images, and in Section 5 we provide several security tests. Section 6 presents the example of a complete cryptosystem combining the proposed *P-Box* with a chaotic *S-Box* based on the logistic map. Finally, the main conclusions are given in Section 7.

2. Permutation with chaotic maps

The possibilities of using chaotic maps for data permutation have been demonstrated by several authors (see, e.g. [16,9,17–19,25]), in particular, Sun et al. [25] recently implemented a permutation process with chaotic series generated by a 2D spatial chaotic map. Although, from a theoretical point of view, their idea is very appealing, we will show that it is hardly applicable to a high-dimensional data encryption calling for a very long computational time. To underline this drawback, we consider a 3D $m \times n \times l$ data array in orthogonal Dekart coordinates with X, Y and Z axes. A basic permutation algorithm based on a chaotic function would have to include the following operations:

1. Choose a convenient chaotic function f: taking into account that the major part of the calculation time is spent looking for the next map value, f must be fast and optimized for real hardware or software implementation. Since the security depends on f and its chaotic properties, a balance between speed and security has to be reached.
2. Permute the data structure in the spirit of Sun et al. [25], the 3D array is first cut in slices of thickness 1 along one axis to then rearrange them in a different order. Starting with the X axis, the data is divided between m YZ separated planes labeled with M_i $(i = 1, 2, \ldots, m)$, thus creating an orderly array $M = (M_1, M_2, \ldots, M_j, \ldots, M_m)$. The chaotic function f is then applied to generate a pseudo-random sequence to shuffle all planes. The initial conditions are used as secret keys. The chaotic sequence indicates with which plane M_j the plane M_i $(i \neq j$ for $\forall i, j \in [1, m])$ is permuted with.
3. Repeat the previous step to create for the XZ and XY planes two sequences N and L of sizes n and l, respectively.

An algorithm with such a structure suffers from two main drawbacks:

(i) The calculations of arrays M, N and L are long and slow, because each value in the arrays must be different from the others in the series. Let us suppose, for example, that we are constructing array M. Using the chaotic function f, we create a long series $S = (S_1, S_2, \ldots, S_m)$; the first value of M (i.e. M_1) is readily available, since $M_1 = S_1$ (if the output of f is in the range $[0, 1]$, each value must be multiplied by m to cover the range $[0, m]$, and rounded to the closer integer number). The problem arises as we go further in calculations of M when the next value S_2 is taken and compared with all previous values of M to verify whether or not it has already appeared and be discarded if so. This process must be repeated for each and all values of S! As the array's dimension becomes bigger, more and more values have to be discarded while the computational time increases very fast, as shown in Fig. 1. Both calculation rates V and R for