Computers and Electrical Engineering 40 (2014) 931–941

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

An image encryption scheme based on constructing large permutation with chaotic sequence ☆

Xuanping Zhang [a,*], Liping Shao [b], Zhongmeng Zhao [a], Zhigang Liang [a]

[a] *Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China*
[b] *School of Computer Science, Shaanxi Normal University, Xi'an 710062, China*

ARTICLE INFO

Article history:
Available online 17 September 2013

ABSTRACT

This paper proposes a chaos-based image encryption scheme with a permutation–diffusion structure. In the proposed scheme, the large permutation with the same size as the plain-image is used to shuffle the positions of image pixels totally. An effective method is also presented to construct the large permutation quickly and easily by combining several small permutations, where small permutations are directly generated using a chaotic map. In the diffusion stage, the pixel is enciphered by *exclusive or* with the previous ciphered pixel and a random number produced by the Logistic map with different initial conditions. Test results and analysis by using several security measures have shown that the proposed scheme is efficient and reliable, and can be applied to real-time image encryption.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The rapid growth of computer networks and the development in digital multimedia processing enable a large number of digital images to be easily transmitted in open network [1]. To protect images from unauthorized access has become a common interest in both research and application. Due to some intrinsic properties of digital image such as bulky data capacity and high correlation among pixels, traditional data encryption techniques may not be suitable for images [2]. Motivated by the chaotic properties such as aperiodicity, high sensitivity to initial conditions and parameters, ergodicity and pseudo-randomness, many researchers have investigated and analyzed various chaos-based image encryption schemes [3].

Matthews [4] first used Logistic map in image encryption and proposed a chaos-based image encryption scheme, then Scharinger [5] presented a Baker map based image encryption algorithm. Fridrich [6] suggested that a chaos-based image encryption scheme should comprise the iteration of permutation and diffusion [7], and Chen [8] used the 3D cat map to encrypt image. Recently, a lot of chaos-based image encryption schemes have been proposed [9–15]. Most of them are based on permutation. These schemes first produce a pseudo-random permutation from chaotic map and then use it to permute images [16].

A typical method for generation of a permutation using one-dimension chaotic map is given in [17]. However, it needs huge computations to generate a large permutation by sorting. Moreover, because of the chaos periodicity caused by finite precision effect [18], chaotic map may produce duplicate elements in chaotic sequence, which increases the difficulty to generate exclusive elements for a large chaotic sequence. In order to avoid the difficulties in generating large permutation, some algorithms use small permutations to shuffle image pixels locally. But this method needs many iterations to achieve an effect equivalent to that of a large permutation.

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.

* Corresponding author. Tel.: +86 02982668645; fax: +86 02982668971.
E-mail addresses: zxp@mail.xjtu.edu.cn (X. Zhang), slpmaster@163.com (L. Shao).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.08.008

In order to provide a reasonable security, a lot of chaos-based image encryption algorithms repeatedly shuffle image pixels by using a Permutation Array (PA) which consists of many different permutations. Since it's quite expensive to generate a large-scale PA directly, the current research is concentrated on how to combine several small PA into a big one under certain constraints [19–22]. Based on Ding's work [19], Yoon [16] proposed an image encryption algorithm by using large PA. However this algorithm requires a large amount of time and storage to construct a PA. Given an image with n pixels, the algorithm will produce a PA containing n-permutations of length n, and requires $O(n^2)$ storages. In practical image encryption, the number of permutations needed is far less than n. Therefore most of the generated permutations are actually unused.

In this paper, a rapid and efficient method for generating large permutation is proposed by introducing the combination operation on permutations. By this operation, a large permutation can be generated by combining several small permutations. Based on this method, an image encryption algorithm is proposed. We also design a diffusion method based on the operations of *exclusive or* and *shift* to increase the security. Experiment results show that the proposed algorithm is efficient and reliable.

The rest of this paper is organized as follows. Section 2 introduces combination operation of permutations, proves some properties of this operation, and presents how to construct the large permutation. Section 3 describes the proposed image encryption algorithm in detail. Section 4 shows results of experiments over various security measures, and conclusion of this paper is given in Section 5.

2. Combination of permutations

Although some techniques have been developed to construct large size PA, they expend a mass of time and storage. Therefore we present a method to generate large permutation by introducing the combination operation, which can produce a single permutation each time.

2.1. Preliminaries

Let $Z_n = \{0,1,2,\ldots,n-1\}$, a permutation π over Z_n, represented as $\pi = (\pi_0,\pi_1,\ldots,\pi_{n-1})$, is a one-to-one self map of set Z_n, where $\pi_i \in Z_n$. For any $i \in Z_n$, $\pi(i) = \pi_i$, which means that i is mapped into π_i by π.

If x and y are two permutations over Z_n, the Hamming distance of x and y is defined by (1), where $\|\|$ means the cardinality of a set.

$$d_H(x,y) = \|\{i|x_i \neq y_i, i \in Z_n\}\| \tag{1}$$

Let S_n denote the set of all $n!$ permutations over Z_n. If the Hamming distance of any two permutations in $C \subseteq S_n$ is at least d, we would call C the permutation array of (n,d), denoted as (n,d)PA.

Suppose C is a (n,d)PA over Z_n of size m, C can be represented by an $m \times n$ matrix in which every row is a permutation over Z_n. We say that C is r-bounded if no element of Z_n appears more than r times in any column of C; C is r-balanced if each element of Z_n appears exactly r times in each column of C; and C is r-separable if it is the disjoint union of r (n,n)PA of size n.

2.2. Combination operation

Let $x = (x_0,x_1,\ldots,x_{m-1})$ and $y = (y_0,y_1,\ldots,y_{n-1})$ be permutations over Z_m and Z_n respectively, the combination of permutations x and y, denoted by $x \otimes y$, is a sequence of mn elements, which is defined by (2), where $\bigsqcup$ is concatenation operator of sequences.

$$x \otimes y = \bigsqcup_{i=0}^{m-1}\bigsqcup_{j=0}^{n-1}(nx_i + y_j) \tag{2}$$

Let $p = x \otimes y = (p_0,p_1,\ldots,p_{mn-1})$, and for any $i \in Z_m$, $j \in Z_n$ and $k = ni + j$, then p_k can be calculated by (3).

$$p_k = nx_i + y_j = nx_{\lfloor k/n \rfloor} + y_{k\%n} \tag{3}$$

Lemma 1. *If x, y and z are permutations, then $(x \otimes y) \otimes z = x \otimes (y \otimes z)$.*

Proof. Let $x = (x_0,x_1,\ldots,x_{m-1})$, $y = (y_0,y_1,\ldots,y_{n-1})$, $z = (z_0,z_1,\ldots,z_{l-1})$, $p = (x \otimes y) \otimes z$ and $p' = x \otimes (y \otimes z)$. Denote $u = x \otimes y$ and $v = y \otimes z$, then $p = u \otimes z$ and $p' = x \otimes v$. For any given $i \in Z_m$, $j \in Z_n$ and $k \in Z_l$, we have

$$\begin{aligned}
u_{ni+j} &= nx_i + y_j \\
v_{lj+k} &= ly_j + z_k \\
p_{nli+lj+k} &= p_{l(ni+j)+k} = lu_{ni+j} + z_k = nlx_i + ly_j + z_k \\
p'_{nli+lj+k} &= p'_{(ln)i+(lj+k)} = nlx_i + v_{lj+k} = nlx_i + ly_j + z_k
\end{aligned}$$

Download English Version:

https://daneshyari.com/en/article/454963

Download Persian Version:

https://daneshyari.com/article/454963

Daneshyari.com