Journal of Security and Safety Technology 安防技术, 2016, 4(1), 1-9
Published Online March 2016 in Hans. http://www.hanspub.org/journal/jsst
http://dx.doi.org/10.12677/jsst.2016.41001

Hans汉斯

Cryptographic Algorithm Based on Digital Black Hole

Jianqiao Zhu[1], Tiankai Sun[1,2], Renwei Zhang[1], Xuanyu Yin[1]

[1]School of Information and Electrical Engineering, Xuzhou Institute of Technology, Xuzhou Jiangsu
[2]Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian Liaoning
Email: strongtiankai@163.com

Received: Mar. 1st, 2016; accepted: Mar. 18th, 2016; published: Mar. 21st, 2016

Abstract

Due to the use of long random key and random output with no statistical relationship between the plaintext, the security of one-time pad plan is very high and cannot be decoded. But the key can be used only once, otherwise it gave attackers a gap from the frequency, and the key is too long to pass or save safely. To the defects of one-time pad, this paper puts forward an algorithm based on the law that any four digit positive integer (Not all the same four figures) will return to a constant number 6174 for up to 7 steps after rearrangement for subtraction, so to the given plaintext and the given key, the ciphertext can be different every time, thus, a string of key can be circularly used for a couple of times which contributes to relieve one-time pad's inherent defects in part.

Keywords

Cryptographic Algorithm, One-Time Pad, Digital Black Hole, Rearrangement Differencing

基于数字黑洞的密码算法

朱剑桥[1]，孙天凯[1,2]，张仁伟[1]，尹轩宇[1]

[1]徐州工程学院信电工程学院，江苏 徐州
[2]大连理工大学电信学部，辽宁 大连
Email: strongtiankai@163.com

文章引用: 朱剑桥, 孙天凯, 张仁伟, 尹轩宇. 基于数字黑洞的密码算法[J]. 安防技术, 2016, 4(1): 1-9.
http://dx.doi.org/10.12677/jsst.2016.41001

收稿日期：2016年3月1日；录用日期：2016年3月18日；发布日期：2016年3月21日

摘　要

传统的一次一密方案由于使用与消息等长的随机密钥，产生与原文没有任何统计关系的随机输出，因此该密码算法安全性很高，加密方案很难破解。但该方案中密钥只能使用一次，不然便给攻击者留下了从频度上攻击的缺口，且密钥太长传递不安全、不利于保存。针对一次一密方案的固有缺陷，本文提出了一种基于黑洞理论的密码算法，该算法基于任意的四位十进制正整数，在至多进行7步重排求差运算后，回归到一个定数的性质，从而实现了对于给定的明文、用确定的密钥加密而随机产生的密文每次都不相同，该方案可使一串密钥循环使用多次，一定程度上缓解了一次一密方案的固有缺陷。

关键词

密码算法，一次一密，数字黑洞，重排求差

1. 引言

一次一密(one-time pad)指在流密码当中使用与消息长度等长的随机密钥，密钥本身只使用一次。具体而言，首先选择一个随机位串作为密钥，然后将明文转变成一个位串，比如使用明文的 ASCII 表示法。最后，逐位计算这两个串的异或值，由此得到的密文很难被破解，因为即使有了足够数量的密文样本，每个字符的出现概率都是均等的，任意字母组合出现的概率也是相等的[1]。一次一密，由于使用与消息等长的随机密钥，产生与原文没有任何统计关系的随机输出，因此一次一密方案很难破解。该方案的主要缺陷是，密钥在传递和分发上存在很大困难[2] [3]。

本文给出的密码算法一定程度上拓展了一次一密算法，使之成为“多次多密”算法，同时给出的密码算法可以通过循环使用较短的密钥对明文加密，解决了密钥过长而带来的密钥在传递、分发和保存上的不安全性因素。

2. 数字黑洞的简介

2.1. 基本概念

黑洞(Black hole)是现代广义相对论中，宇宙空间内存在的一种超高密度天体，由于类似热力学上完全不反射光线的黑体，故名为黑洞。黑洞产生的引力场极为强劲，以至于任何物质和辐射在进入到黑洞的一个事件视界(临界点)内，便再无力逃脱，甚至目前已知的传播速度最快的光(电磁波)也逃逸不出。数学中借用这个词，指的是某种运算，这种运算一般限定从某些整数出发，(通常为十进制正整数且组成该数的各位数字不全相同，下文中“整数”一词若不加声明均指受该限定的整数)反复迭代后结果必然落入某一个反复出现的确定的整数或周期性循环出现的若干个相互关联的整数的现象，由于产生的效应就像被黑洞吞噬了一样再也无法逃脱，通常这种现象被称为数字黑洞[4] [5]。

实现数字黑洞现象最简单常见的运算描述如下：首先取任意三位数及三位以上的整数 N。然后将组成该整数的所有位上的数字，按数值大小从高位到低位降序重新排列后得到一个最大的整数 Max，再将组成该整数的这些数字，按数值大小从高位到低位升序重新排列后得到一个最小的整数 Min，将重新排列后得到的最大的整数 Max 减去重新排列后得到的最小的整数 Min 得到它们的差 M，其中上述操作称为重排求差操作。若将 M 视为 N 再次进行重排求差操作，最终总会得某一个反复出现的确定的整数或周期

性循环出现的若干个相互关联的整数，这些数又被称为黑洞数或陷阱数、黑洞数组或陷阱数组。将某个整数经过若干步重排求差后最终得到黑洞数的现象被称为重排求差黑洞或卡普雷卡尔黑洞。

2.2. 重排求差数字黑洞举例

2.2.1. 三位数的重排求差黑洞 495

任意一个三位整数，个位、十位、百位数字不全相同，例如不允许 111，222，333……的出现，百位、十位数字若为 0 要将其视为三位数。将这个三位数进行若干步重排求差操作，最终总会得到 495 这个数字。具体举例如下：

任意的三位整数 N = 352，排列得最大数 Max = 532，最小数 Min = 235，求差得 M = 297；将 M 视为新的 N，得 Max = 972、Min = 279，求差得 M = 693；将 M 视为新的 N，得 Max = 963、Min = 369，求差得 M = 594；将 M 视为新的 N，得 Max = 954、Min = 459，求差得 M = 495；将 M 视为新的 N，得 Max = 954、Min = 459，求差的 M = 495……由此可见该运算一旦得到 495 便会一直重复“954 − 459 = 495”这一步骤。

2.2.2. 四位数的重排求差黑洞 6174

任意一个四位整数，个位、十位、百位、千位数字不全相同，例如不允许 1111，2222，3333……的出现，千位、百位、十位数字若为 0 要将其视为四位数。将这个四位数进行若干步重排求差操作，最终总会得到 6174 这个数字。具体举例如下

任意的四位整数 N = 3109，排列得最大数 Max = 9310，最小数 Min = 0139，求差得 M = 9171；将 M 视为新的 N，得 Max = 9711、Min = 1179，求差得 M = 8532；将 M 视为新的 N，得 Max = 8532、Min = 2358，求差得 M = 6174；将 M 视为新的 N，得 Max = 7641、Min = 1467，求差得 M = 6174；将 M 视为新的 N，得 Max = 7641、Min = 1467，求差的 M = 6174……由此可见该运算一旦得到 6174 便会一直重复“7641 − 1467 = 6174”这一步骤。

2.2.3. N 位数的重排求差黑洞

任意 N 位数都会类似三位、四位数那样归敛(1、2 位数无意义)。三位数归敛到唯一一个数 495；四位数归敛到唯一一个数 6174；七位数归敛到唯一一个数组(8 个 7 位数组成的循环数组，称为归敛组)；其它每个位数的数归敛结果分别有若干个，归敛数和归敛组兼而有之(如 14 位数共有 9×10 的 13 次方个数，归敛结果有 6 个归敛数，21 个归敛组)。以上提到的所有归敛结果(包括一个数字、一个数组或兼有)称为“卡普雷卡尔常数”。任意 N 位数的归敛结果都“隐藏”在这 N 位数中，卡普雷卡尔运算只是找出它们而不是新造成它们。

“卡普雷卡尔常数”还有一些奇妙的性质，例如：“卡普雷卡尔常数”中的所有的数都是模 9 数(即都能被 9 整除以及其全部数字之和也是 9 的倍数)；并且一旦进入归敛结果，继续卡普雷卡尔运算就在归敛结果反复循环，再也“逃”不出去。在此本文只对四位数的重排求差数字黑洞进行分析，且根据四位数的重排求差数字黑洞的性质提出一种密码算法[6]-[10]。

2.3. 四位数字黑洞的重要性质及其证明

定理 1 任意的一个四位且各数位上数字不全相同的十进制正整数，将其进行重排求差操作，至多不超过 7 步就必然得到 6174。

该定理容易通过穷举法给出机器证明，在此不赘述。下面给出简化的列举证明：

对于任意的四位整数 N，设 a、b、c、d 是组成 N 的数字，且 $a \geq b \geq c \geq d$，又因为它们不全相等，所以上式中的等号不能同时成立，将对 N 的重排求差操作记为 T(N)。下面计算 T(N)：

$$\mathrm{Max} = 1000a + 100b + 10c + d$$
$$\mathrm{Min} = 1000d + 100c + 10b + a$$
$$T(N) = \mathrm{Max} - \mathrm{Min} = 1000(a-d) + 100(b-c) + 10(c-b) + (d-a) = 999(a-d) + 90(b-c)$$

我们注意到 T(N)仅依赖于(a-d)与(b-c)，又因为数字 a、b、c、d 不全相等，因此由 $a \geq b \geq c \geq d$ 可推出：$(a-d) > 0$ 而 $(b-c) \geq 0$；此外 b、c 在 a 与 d 之间，所以 $(a-d) \geq (b-c)$，这就意味着(a-d)可以取 1、2、⋯、9 九个值，并且如果它取这个集合的某个值 n，(b-c)只能取小于 n 的值，至多取 n。

把 $(a-d) = 1, (a-d) = 2, \cdots, (a-d) = 9$ 的情况下(b-c)所可能取值的个数加起来，我们就得到 2 + 3 + 4 + ⋯ + 10 = 54，这就是 T(N)所可能取的值的个数。在这 54 个可能值中，又有一部分是数码相同仅仅是数位不同的值，这些数值再变换 T(N)中都对应相同的值(数学上称这两个数等价)，剔除等价的因数，在 T(M)的 54 个可能值中，只有 30 个是不等价的，如表 1 所示。

对于这 30 个数逐个地用上述法则把它换成最大数与最小数的差，至多 6 步就出现 6174 这个数，再加上由 N 重排求差后得到以上 30 个不等价的数的步骤，则刚好 7 步便得到 6174 这个数。证毕。

3. 密码算法设计方案

3.1. 可行性分析与准备工作

根据前面的分析，任意的一个四位且各数位上数字不全相同的十进制正整数，将其进行重排求差操作，至多不超过 7 步就必然得到 6174。如果再将那些各数位上数字全都相同的四位正整数的回归 6174 的步数记作 0 步，且将不足四位的正整数在高位补 0，并将其视为四位数，那么我们就建立了一个全体四位正整数(共 10,000 个数字)到 0~7 (共 8 个数字)的映射，如图 1 所示。

根据此映射可以设计出二进制流明文到十进制流密文的加密，以及十进制流密文到二进制流明文的解密。通过计算对按降序排列的不等价的四位正整数，穷举计算可得任意四位正整数重排求差后回归 6174 的步数分别为 0、1、2、3、4、5、6、7 的数字的个数，如表 2 所示。

Table 1. 30 inequivalent number
表 1. 不等价的 30 个数

9990	9981	9972	9963	9954
9810	9711	9621	9531	9441
8820	8730	8721	8640	8622
8550	8532	8442	7731	7641
7632	7551	7533	7443	6642
6552	6543	6444	5553	5544

Table 2. Statistic of recurrence step and total number
表 2. 回归步数与数字个数统计

回归所需的步数	数字的个数
0	9
1	20
2	34
3	140
4	129
5	113
6	153
7	116

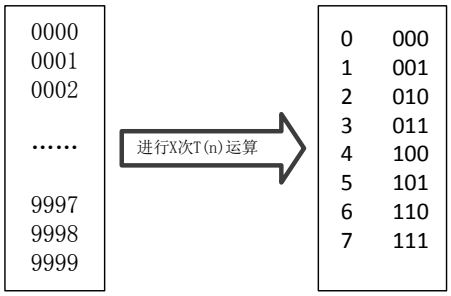

Figure 1. Mapping from 0000 - 9999 to 0 - 7
图 1. 从 0000~9999 到 0~7 的映射

由表 2 可见除了回归步数在 0、1、2 之外，数字的个数分布大体均衡。这时只需将所有回归步数为 0、1、2、……7 的数字(包括那些组成数字都相同而所在的位置不相同的等价的四位正整数)，分别存于“数表 0”、“数表 1”、……、“数表 7”中，便构造出了一个密码本，供后续加密算法使用。

3.2. 加密算法的设计方案

3.2.1. 加密算法思想

基于计算机世界中所有的信息均以 0、1 形式表示，可以先将二进制流从头至尾每三位一组，最后的一组二进制数若不足三位则不对其进行加密操作，再从每组二进制数 Xi 对应的数表中随机选取一个数 N，最后与一个随机产生的秘钥 Key 做模 10000D 相加得密文 Si。对每一组二进制数 X0~Xn 进行上述运算后，可以得到一串对应的密文 S0~Sn，从而对任意的文件进行加密。

值得说明的是，这里的秘钥每完成一次对一组二进制流的加密则丢弃，需要重新随机生成秘钥，或者提前制定一串安全意义上足够长的密钥，根据事先制定的规则，在加密过程中每加密一组数据使用一段秘钥，秘钥越长安全性越高，这样便可以在长时间内都使用相同的秘钥，避免了秘钥交换的不便性。

由于加密算法随机选取数字作为二进制流明文的替换，以及密钥也是随机选取或人为制定的，那么即使对相同的明文进行加密也可以得到不同的密文。

3.2.2. 加密算法描述

第一步：将待加密文件按某种规则转换成二进制流，值得说明的是在这一步可以自行制定转换规则，以进一步增强加密算法的安全性；

第二步：将二进制流从头至尾每三位二进制一组，对二进制流分组，若最后一组不足三位，不对其进行加密操作，加密过程中始终不变；

第三步：从第一组开始，逐一从对应的数表中随机选取一个数 N；

第四步：将 N 与随机产生的一段密钥 K(或事先制定的一段秘钥 K)作模 10000D 加；

第五步：重复第三步、第四步，直到对所有的分组都加密完毕，最终得到一串十进制数字流密文。如图 2 所示。

3.2.3. 加密算法举例

若要将字符 A 加密，事先随机产生的秘钥为：(3452)(3125)(9849)……

1) 按其对应的 ASCII 码转为二进制流为：01000001；
2) 按三位一组进行分组得：(010)(000)(01)(最后的分组不足三位，加密过程中始终保持不变)；
3) 根据每个分组在其对应的数表中随机选取的数分别是：4637、5555；
4) 将随机选取的数与密钥进行模 10000D 运算加密得：(8089)(8680)(01)。

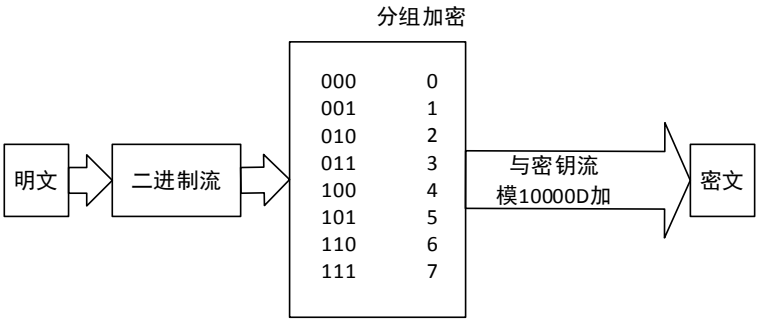

Figure 2. Graphic of encryption algorithm
图 2. 加密算法图示

3.3. 解密算法的设计方案

3.3.1. 解密算法思想

解密过程十分简单，首先只需将每组密文与对应的每组密钥进行模 10000D 减运算，得到一个四位整数；若这个四位整数的各个数位的数码都相同，则二进制流明文为 000；若这个四位整数的数码不全相同，则将这四位整数进行重排求差运算，直到得到 6174 为止，并记录这个四位整数回归到 6174 的步数 t，再将 t 转换为三位的二进制数(不足三位在其高位补 0)，最终操作完所有的分组，得到完整的二进制流明文；最后按照加密时将文件转换成二进制流文件方案，将二进制流文件转换成相应格式的文件。

3.3.2. 解密算法描述

第一步：将十进制密文数字流四位一组进行分组；

第二步：从第一组开始，逐一与对应的密钥进行模 10000D 减运算，对结果进行重排求差操作，直到结果第一次出现 6174，重排求差操作的次数的二进制表示即为二进制明文；(或者使用更高效率的方法：将第一次重排求差后得到的数字，从高位到低位由大到小重排，所得的数必然是表 1 中的 30 个不等价的数，而这些数的回归步数是确定的，事先计算好用实际只需查看即可。)

第三步：重复第二步直到对所有的十进制密文数字流解密完毕，若结尾的分组不足四位则不做任何解密操作，整个解密过程中保持不变，最终得到一串二进制数字流明文；

第四步：将二进制流明文转换成文件即可得到原始文件。如图 3 所示。

3.3.3. 解密算法举例

若得到密文：8089868001，已知秘钥为：(3452)(3125)(9849)……

1) 按四位一组进行分组得：(8089)(8680)(01)；

2) 计算每个分组与其对应的密钥模 10000D 减的结果得：4637、5555、01，并且求出这两个数重排求差操作后回归 6174 的运算次数以二进制表示，得：010、000、01；

3) 最终得到二进制流明文：01000001

4) 查找 ASCII 码表转为字符：A。

4. 利弊分析及优化思路

4.1. 算法优点分析

其一，加密、解密时没有任何复杂的数学计算；加密时的主要运算只是模 10000D 加法，而且用来与秘钥相加的数均通过在对应表中随机选择得到；而解密时主要的运算是重排四位数和减法运算，由于

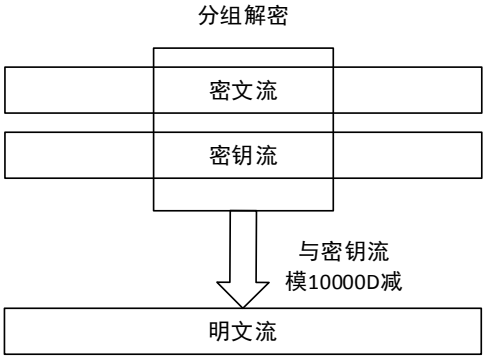

Figure 3. Graphic of decryption algorithm
图 3. 解密算法图示

排序的数字仅有四个，并且排序的次数也不多(至多只需排序 14 次)，甚至用高效的方法只需进行两次重排一次求差，即可直接查到回归步数，所以解密的效率也很高。

其二，在加密的过程中，由于用来与秘钥相加的数均是通过在对应表中随机选择得到的，而且秘钥本身可以是人为制定，也可以在加密每个二进制分组时随机选取；这样可以实现对于同样的明文同样的秘钥进行加密得到的密文绝对不同，那么就避免了攻击者从密文信息的频率上破解明文，这种加密方案是不可破解的、安全性很高。

其三，该加密方案，可以看成是分组加密与一次一密加密方案的结合与改进，从而使得一次一密不停留在“一次”上，同样的密钥加密过一次后，密钥的安全性并不降低，实现了密钥的重复使用，缓解了一次一密方案中密钥发送困难的问题。由于一段相同的密钥与一段相同的明文加密后得到密文相同的概率几乎为 0，因此可以循环使用同一段密钥对较长的明文加密，从而减短密钥的长度。

4.2. 潜在的安全性威胁分析

由于 000、001、010 这些二进制组对应的数表空间并不大(具体参见表 2)，可能给攻击者从频度方面破解留了个缺口，因此二进制流明文中序列“000”、“001”、“010”的出现频率不宜过多。此外，该加密方案将一段明文转换成密文后的拓展比较大，增加了传输时间。

4.3. 优化思路

对于二进制序列“000”、“001”、“010”所对应的数表空间并不大这一固有缺陷，笔者给出两种解决方案。方案一：在将文件转化成二进制流明文时，采用其他的算法使二进制序列“000”、“001”、“010”出现的频率相对较低，达到在整体加密算法中各个序列出现的频度相对均衡。方案二：对二进制流明文加密时在连续集中出现二进制序列“000”、“001”、“010”处，随机地加入约定的“噪声序列”，在解密时将“噪声序列”删除，同样也可以使得在整体加密算法中各个序列出现的频度相对均衡。

5. 启示及结论

5.1. 启示

在此抽象出本文给出的密码算法核心思想，可以得出实现对相同的明文、相同的密钥每次加密所得的密文都不相同，即“多次多密”的抽象实现方法。

首先将明文分组，分组后得到的每组序列都是一个确定的已知的有限集合 A:{a1, a2, ……, an}中的

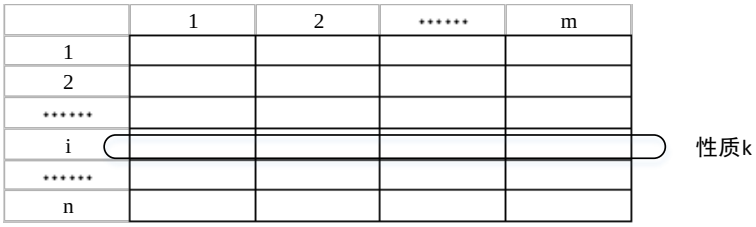

Figure 4. Algorithm's core concept
图 4. 算法核心思想图示

元素，集合 A 中的每个元素又与某个确定的已知的有限集合形成一对一关系，例如集合 A 中的元素 ai 对应于集合 Bi:{b1, b2, ……, bm}。这些与集合 A 中全体元素形成一一对应关系的集合是集合 C:{B1, B2, ……, Bm}中的元素，而对于集合 C 中任意的 Bi，它的全体元素都有某种相同的性质，我们不妨将这种性质称作集合 Bi 的性质，对于集合 C 中任意的两个元素 Bi、Bj，它们的性质都不相同，且 Bi 中所有的元素都不具备集合 Bj 的性质，即对于任意的 Bi 中的任意元素 bi 都可以根据 bi 的性质找出 bi 所属的一个唯一的集合 Bi。

换言之，根据分组序列的类型，可以唯一确定某已知的有限的二维表中的某一行。如图 4 所示。

该二维表中任意一行的所有元素都有某种相同的性质，不妨将这种相同的性质称为行性质，而二维表中所有的行性质又都不相同，且任意一行中的所有的元素都不具备其它的行性质，即根据任意一行的任意元素的性质都可以确定该元素所在的唯一的行。

于是可以将集合 C 中某个 Bi 的全体元素都可视为等价的，即第 i 行中的所有元素都是等价的，由于它们都有具有相同的性质 k 而仅仅是表现形式不同，则可以随机选取该行的某个元素 bi 作为对某个分组序列的替换，再将 bi 与密钥运算即可得到密文，至此加密过程完毕。

解密时只需将密文与密钥通过解密算法求出 bi，再根据 bi 的性质可以确定唯一的分组序列，解密时甚至无需解出 bi 的具体值只需得出 bi 所属集合 Bi 的性质即可。将解出的所有分组序列连接即可得到明文。

5.2. 结论

本文给出了一种基于四位十进制正整数重排求差数字黑洞的性质，从而实现对给定的明文和某确定的密钥加密所得到的密文每次都不相同(具有抽象可能性的相同)。一定程度上改良拓展了“一次一密”算法，使之成为“多次多密”算法。最后在文章结尾介绍了实现“多次多密”算法的一般思路。

基金项目

徐州市科技计划项目(XM13B126)徐州工程学院青年基金(XKY2015305)。

参考文献 (References)

[1] 王勇. 一次一密的安全性与新保密体制[J]. 信息网络安全, 2004(43): 41-43.

[2] 范畅, 茹鹏. 非线性一次一密(t,n)门限秘密共享方案[J]. 计算机应用, 2013, 33(9): 2536-2539.

[3] 潘江游, 杨理. 基于一次一密的量子身份识别方案[J]. 中国科学院研究生院学报, 2012(3): 277-281.

[4] 黄振国. 黑洞数的性质与它神奇的衍生法[J]. 广西大学梧州分校学报, 2004(1): 62-64.

[5] 杨之, 张忠辅. 角谷猜想和黑洞数问题的图论表示[J]. 自然杂志, 1988(6): 453-456.

[6] 王传彪. 黑洞研究中的数学方法[D]: [硕士学位论文]. 吉林: 吉林大学, 2011.

[7] 王子成, 赵晓航, 王宏. 基于 DNA 密码的一次一密加密算法[J]. 计算机工程与应用, 2014(15): 97-100.

[8] 田国胜, 杨昆, 张民. 类似一次一密的复合混沌音频隐藏方案[J]. 中国安全科学学报, 2009(4): 97-101.

[9] Zhang, Y.A. and Feng, D.G. (2005) A Practial One-Time Pad Like Block Cipher Scheme. **2**, 101-104.

[10] Huang, Y.H., Hu, A.Q. and Song, Y.B. (2004) A Study on the One-Time Pad Algorithm and Its Implemetation. **3**, 287-290.