Open Journal of Circuits and Systems 电路与系统, 2013, 2, 39-44
http://dx.doi.org/10.12677/ojcs.2013.23007 Published Online September 2013 (http://www.hanspub.org/journal/ojcs.html)

Realization of Encryption on FPGA Based on Chaotic Signals Generated by Logistic Equation

Lanjie Shi, Shijian Gao, Xi Huang, Haojie Yuan, Tiege Zhou

College of Information Technical Science, Nankai University, Tianjin
Email: shilanjie@gmail.com

Received: Mar. 19th, 2013; revised: May 1st, 2013; accepted: May 18th, 2013

Abstract: Chaotic systems can produce effective pseudo-random sequences. Chaotic sequences are suitable for stream ciphers because they are sensitive to the initial value and unpredictable. Information can be encrypted with chaotic sequences generated by the Logistic equation, which is easy to be realized on hardware. The theory has been testified by the experimental result on hardware FPGA. In the end, the paper introduced the method to optimize the system.

Keywords: Logistic Equation; Chaotic Signal; Encryption and Decryption; FPGA

基于 FPGA 的 Logistic 方程混沌信号加密实现

石兰洁，高诗简，黄　曦，原豪杰，周铁戈

南开大学信息技术科学学院，天津
Email: shilanjie@gmail.com

收稿日期：2013 年 3 月 19 日；修回日期：2013 年 5 月 1 日；录用日期：2013 年 5 月 18 日

摘　要：混沌系统能产生具有对初值敏感、难以预测的性能良好的伪随机序列，很适于序列密码。采用逻辑斯特(Logistic)方程生成混沌序列加密信息，方法简单，硬件上便于实现。然后设计具体硬件实现方法，在现场可编程门阵列(FPGA)上实验，结果得到了验证。最后，针对 Logistic 方程加密的缺点，本文提出了优化方法。

关键词：Logistic 方程；混沌信号；加密；FPGA

1. 引言

序列密码具有实现简单、便于硬件实施、加解密处理速度快、传播错误极少等优点，因此在实际应用中比如无线通信、外交通信有极大的优势。1949 年香农(Shannon)证明了只有一次一密的密码体制是绝对安全的[1]。如果序列密码所使用的是真正随机的、与消息流长度相同的密钥流，则此时的序列密码就是一次一密的密码体制。若能以一种方式产生一随机序列(密钥流)，这一序列由密钥所确定，则利用这样的序列就可以进行加密。

19 世纪 80 年代末，英国数学家 Matthews 第一个提出了混沌加密方法。他提出利用逻辑斯特(Logistic)变形映射产生密钥流来加密信息的方法[2]。简单确定的系统不仅产生简单确定的行为，还可以产生貌似随机的不确定行为，即混沌行为。混沌信号的表现形式非常复杂，难于分析[3]。它具有初值敏感性、类噪声、长期不可预测性等特点，但是混沌系统本身又是确定的，由方程、参数所完全决定，是确定性非线形系统产生的不确定的信号，其状态完全可以重现。这些性质使得混沌信号具有长期不可预测性和很强的抗截

获能力及设计的简单性。所以将混沌掩盖引入通信领域尤其是运用于通信的加密保护具有极其广阔的前景和巨大的潜力。

传统信息加密如 DES 分组密码体制因其密钥长度只有 56 位，故其加解密耗时非常短，但正因为其密钥长度只有 56 位，这种加密不安全。而 RSA 公钥密码体制 RSA 作为高强度的非对称(公钥)数据加密标准，其密钥长度较之 DES 要大得多。但也因为其密钥长度过于长，而且采用公钥加密，加解密较为耗时，效率很低[4]。

与传统信息加密相比，混沌加密算法具有代价小、实现简单、稳定性高等优势。

混沌加密算法通过循环产生密钥，在循环过程中设定的变量很少，则占用的空间也很少。混沌迭代产生的密钥既可以用来加密也可以用来解密。其加密和解密过程算法大体一样，这样大大减小了设计的难度。而且混沌迭代方程很简单，在软件和硬件上实施很方便。混沌信号的产生由方程、参数严格控制，之后的异或计算也是有严格的规则。加密过程步骤简单，这样可以减少加密过程中出错，大大提高了稳定性。

1991 年，Wheeler 指出，由于有限精度效应[5]，混沌系统会出现短周期行为，使得安全性降低，不适合实际应用。之后陆续有人提出了各种算法。

M. S. Baptista 在 1998 年提出了一种基于搜索机制的混沌密码算法[6]，但该算法存在运算速度慢、密文分布不均匀的缺陷。其它算法如 Alvarez 型算法[7]是一种对称分组混沌密码算法，它将每组明文加密为由三个部分构成的密文分组，且分组长度可变。这种算法密文有效率低，加密速度也较慢，因此该算法不适合在明文量大、对加密速度要求高的情况下使用[8]。而且这些算法中，运用了其它混沌生成器，将几种常用的混沌序列生成器进行了比较分析，发现 Logistic 映射最简单，具有很强的初值敏感性，产生的序列随机性好，但正是由于它的形式简单和使用广泛，所以不安全，易于攻击[9]。

本文选择 Logistic 方程生成密钥，着重介绍了在 FPGA 中 Logistic 方程的简化算法，以便适合硬件计算，加快速度，减少空间，然后在 FPGA 上得到了验证。为了减弱有限精度效应，增加安全性，在生成密钥的过程中增加了微小变动，使得既满足设计的简单性又增强了安全性。

2. 方法及实现

2.1. Logistic 加密基本方法及实现

根据逻辑斯特(Logistic)方程：

$X_{n+1}=\lambda X_n(1-X_n)\left(X_n\in[0,1];\lambda\in[0,4]\right)$ 得一系列的数据。在迭代过程中，我们可以通过控制条件来进入混沌状态，进而得到混沌信号。当 $3.5699456<\lambda\le 4$ 时，迭代结果是无穷个不同的值，表现了极大的随机性，系统进入混沌状态[10]。

如图 1 所示，利用 Logistic 方程进入混沌状态得到的一系列数据来覆盖原信息。由 $X_{n+1}=\lambda X_n(1-X_n)$，我们可以得到一系列的看起来杂乱无章的但又是由确定性系统产生的信号即密码序列。利用此信号的不确定性可以和信号的数据进行异或计算，将信息转换成另一种形式，完成加密，从而达到保护信息的目的。

信息的解密又是另一个独立的过程，我们先用同样的参数和初始值，迭代同样次数，产生相同的混沌信号即密码序列，再将密文同密码序列经过异或计算得到明文。

根据 Logistic 映射

$$X_{n+1}=\lambda X_n(1-X_n)\left(X_n\in[0,1];\lambda\in[0,4]\right) \quad (1)$$

由公式可以看出，计算时采用的都是浮点数，但在实际的通信系统中，浮点数运算内存需求大，而且运算速度慢，几乎不可能实现数字语音信号的实时加密。为此我们利用等效变换使混沌序列的浮点迭代过程变为整点迭代过程。在本次研究中，我们以一个字节(8 位)为单位对数据进行加密。令

$$Y_n=2^8\times X_n=256X_n \quad (2)$$

带入(1)式后，得到

$$Y_{n+1}=\lambda Y_n(256-Y_n)/256 \quad (3)$$

设定 $\lambda=3.9$，由于浮点数在 FPGA 中不好计算，我们先将 3.9 扩大 256 倍得到 998，然后再缩小 256 倍。带入(3)式后得到

$$Y_{n+1}=998Y_n(256-Y_n)/256/256 \quad (4)$$

根据 FPGA 的性质，我们进行了算法优化。定义

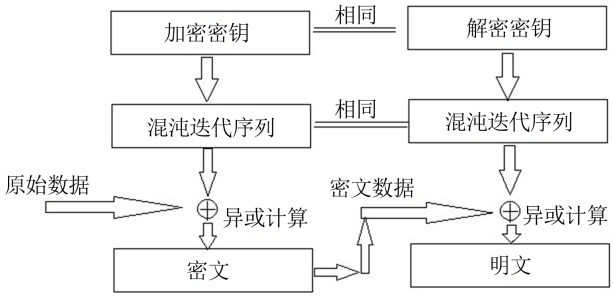

Figure 1. Flow chart: chaotic encryption and decryption algorithms
图 1. 混沌加密解密算法流程图

8 位数据 Y_1，计算 $998\times Y_1\times(256-Y_1)$。998 是 10 位数据，得到的是 $10+8+8=26$ 位数据。又由于 $Y_1\times(256-Y_1)\le 128\times 128=16384$，则

$$998\times Y_1\times(256-Y_1)\le 998\times 16384$$

又 $998\times 16384=16351232<2^{24}=16777216$，说明得到的 26 位数据前 2 位是 0，因此结果中我们不取前 2 位，只关心后 24 位。根据 FPGA 的性质，将 $998Y_n(256-Y_n)$ 除以 256×256 时，只需将结果的后 16 位去掉，即我们取得结果的高 8 位。综上可得 Y_2 为 $998\times Y_1\times(256-Y_1)$ 的 23 位到 16 位，则得到了 8 位数据的混沌信号。

为了实现设计的思路，我们生成一些原始数据。为此，我们设计了一个 8 位的计数器。用计数器得到的数据作为原始数据和 8 位混沌信号进行异或计算，得到加密后的数据。为了便于观察，我们利用 FPGA 上的开发板上的数码管将数据显示出来[11]。

在解密的过程中，我们将在同样条件下产生一样的混沌信号，再与加密的数据进行异或计算，得到还原的数据，即明文。图 2 是 FPGA 设计的顶层文件原理图。int_div 是分频文件，将 FPGA 内的高频时钟信号降下来。jiami 模块主要是生成原始数据、混沌信号，然后完成加密过程。其中 QOUT1 是原始数据，QOUT2 是混沌信号，MOUT 是加密信号输出。其它的模块主要是扫描显示，便于观察。图 3 是显示的现象，我们将 8 位数据分成 2 个 4 位数据，然后用数码管显示，即转换成 16 进制显示。其中，左边 2 个是原始数据，中间 2 个是混沌信号，右边 2 个是加密后的信号。图中，显示了原始数据为 01H 和 08H 的情况，即 01H^9FH = 9EH；08H^86H = 8EH。图 4 中，显示的是解密情况，左边 2 个是加密后的数据，中间 2 个是混沌信号，右边 2 个是解密后的信号。图中，显示了加密数据为 9EH 和 8EH 的情况，即 9EH^9FH = 01H；

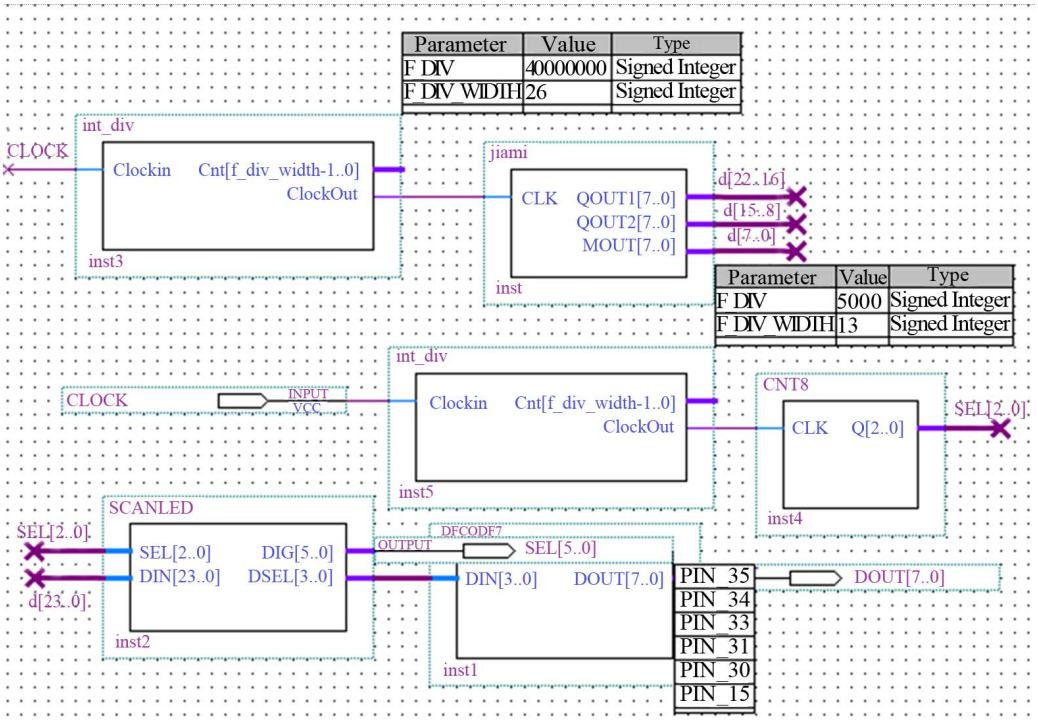

Figure 2. Top-level entity on FPGA
图 2. FPGA 顶层文件图

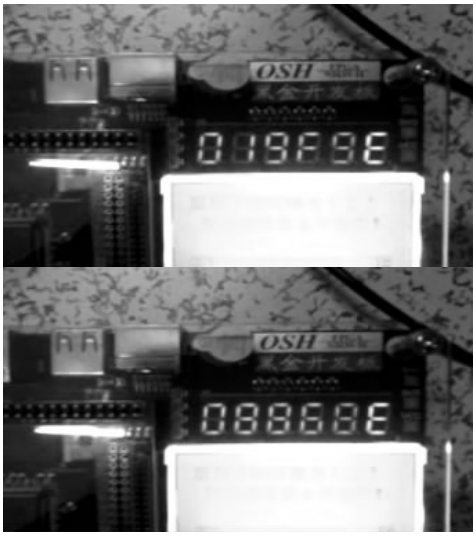

Figure 3. Encrypted data on LED
图 3. 数码管显示加密数据

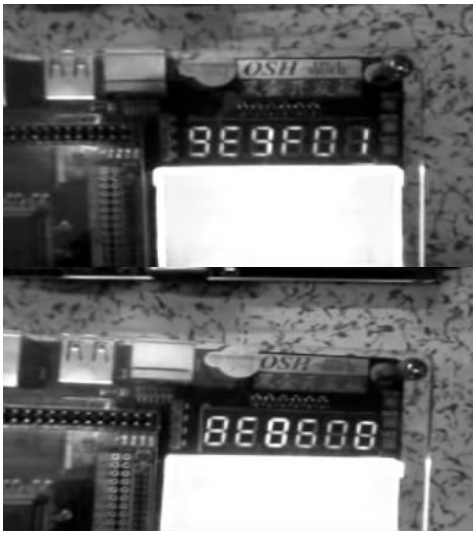

Figure 4. Decrypted data on LED
图 4. 数码管显示解密数据

8EH^86H = 08H。有上图可知，加密解密所用的混沌信号是一样的，说明可以利用这一点进行信息的解密。

2.2. Logistic 加密方法改进及实验验证

由于有限精度效应[5]，在加密大量数据时，需要产生大量混沌信号，这些信号出现短周期行为，混沌效果减弱。在上述计算时我们采用的 8 位数据，迭代过程中，数据会严重丢失，使得混沌信号出现周期情况。利用矩阵实验室计算软件(matlab)生成大量的 8 位混沌信号，如图 5 所示，每隔一段时间会有一段的重复。

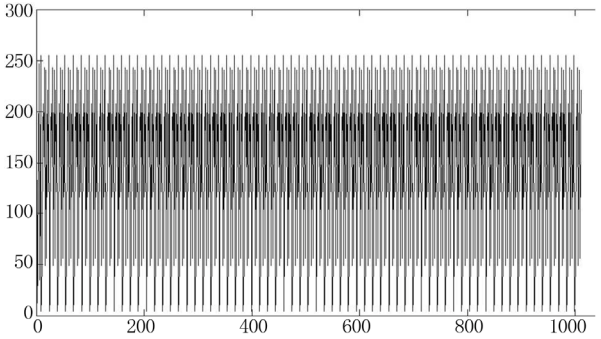

Figure 5. large number chaotic signals before improvement
图 5. 未改进方案时大量混沌信号变化情况

为此，我们设计了改进方案。在生成混沌信号的过程中，我们设定 2 列混沌信号。一列混沌信号 X_n 随个数依次迭代产生，另一列混沌信号 Y_n 每隔 m 个迭代产生。序列号对 m 取模为 0 时 Y_n 迭代，并且计算 $X_n = (X_n + Y_n)\%256$，由这种方法改变迭代的值来破坏周期性。这种改进实现起来比较简单，便于实际应用。如图 6 是改进后产生的混沌信号，可见混沌效果得到了加强。

为了验证上述方法的可行性，我们在 FPGA 上进行了验证。我们选用 Cyclone Ⅱ 系列的 EP2C8Q208C 的 FPGA 芯片，利用 640×480*VGA* 显示接口，连接在计算机显示屏上。先显示了原始图片[11]，如图 7。利用上述未改进方法加密了此图片，如图 8 所示。图 9 是 FPGA 中加密图片的顶层文件原理图。图中，p1 中存储原始图片的数据。Jiami2 是加密数据模块，然后将加密的数据经过 vga640480 显示在屏幕上[11]。由图 8 可见，图片得到了加密，但是由于有限精度效应，还能看出原始图片的轮廓。因此，我们又按照上述方法进行了改进。

由于 vga 显示图片是 3 位数据，我们将 Logistic 方程中的浮点数扩大了 8 倍，便于计算。在加密模块里，有 2 个 Logistic 方程。第一个方程随着数据的个数迭代，第二个方程每 m 个数据迭代一次。图 10 是 $m = 4$ 的情况，可见加密效果得到了加强。图 11 和图 12 是 $m = 2$ 和 $m = 1$ 的情况，可见加密效果得到明显了加强。由此可得，改进 Logistic 方程可以达到很好的加密效果，而且便于在硬件 FPGA 上实现，说明有着很大的实际价值。

3. 结束语

本文介绍了基于 Logistic 方程的混沌加密算法及

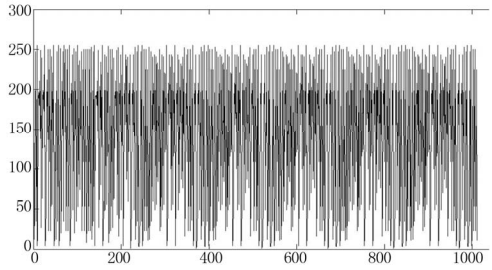

Figure 6. A large number of chaotic signals after improvement
图 6. 改进方案后大量混沌信号变化情况

Figure 7. Original picture
图 7. 原始图片

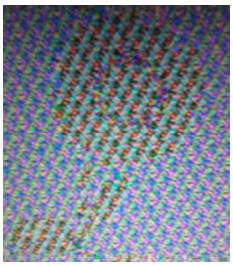

Figure 8. Encrypted pictures before improvement
图 8. 未改进方法加密图片

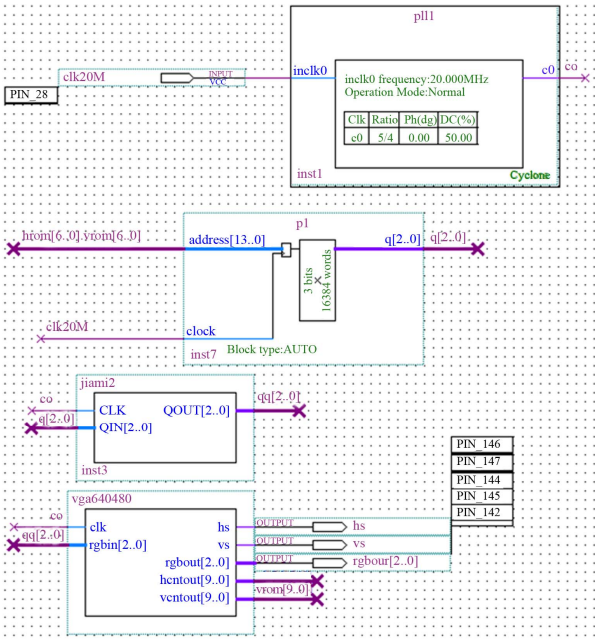

Figure 9. Schematic of encrypted pictures
图 9. 加密图片原理图

Figure 10. Encrypted pictures after improvement (m=4)
图 10. 改进方法(m = 4)时加密图片

Figure 11. Encrypted pictures after improvement (m=2)
图 11. 改进方法(m = 2)时加密图片

Figure 12. Encrypted pictures after improvement (m=1)
图 12 改进方法(m = 1)时加密图片

其实验设计，给出了计算原理和改进方案，并在 FPGA 上得到了实验验证。将混沌信号运用于信息加密领域具有极高的实用价值，有着广泛的应用前景。混沌信号产生的多元化及复杂化也为密码的可靠性及安全性的提高提供了极大的发展空间。

参考文献 (References)

[1] C. E. Shannon. Communication theory of secrecy systems. Bell System Technical Journal, 1949, 28(4): 656-715.
[2] R. Matthews. ON the derivation of a “Chaotic” encryption algorithm. Cryptologia, 1989, 13(1): 29-42.
[3] 张慧源, 禹思敏, 龚大刚. 基于混沌加密的嵌入式通信系统的研究[J]. 微计算机信息, 2005, 32: 31-32.
[4] 李凌昊. 浅谈传统加密原理优缺点及混沌原理在密码学中的作用[J]. 计算机光盘软件与应用, 2011, 21: 93.
[5] D. D. Wheeler, R. A. J. Matthews. Supercomputer investigations of a chaotic encryption algorithm. Cryptologia, 1991, 15(2): 140-152.
[6] M. S. Baptista. Cryptography with chaos. Physics Letters A, 1998, 240(1-2): 50-54.

[7] E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez and A. Marcano. New approach to chaotic encryption. Physics Letters A, 1999, 263(4-6): 373-375.
[8] 刘金梅, 丘水生. 混沌系统在密码学中的应用现状及展望[J]. 计算机工程与应用, 2008, 44(14): 5-12.
[9] 赵玉霞, 樊景博. 几种混沌序列生成器的比较分析[J]. 现代电子技术, 2010, 33(10): 43-45.
[10] 韩风英, 朱从旭. 基于 Logistic 映射混沌加密算法的研究[J]. 长沙航空职业技术学院学报, 2007, 7(1): 30-33.
[11] 孟庆斌, 司敏山. EDA 实验教程[M]. 天津: 南开大学出版社, 2011: 108-126, 232-251.