基于多线程并行混淆与扩散的实时混沌视频加密

Dong Jiang[a、b、*]、Zhen Yuan[a]、Wen-xin Li[a]、Liang-liang Lu[c、d、e、*]

[a]安徽大学互联网学院，合肥，230039，中国[b]安徽大学农业生态大数据分析与应用国家工程研究中心，合肥，230601，中国[c]江苏省光电子技术重点实验室，南京师范大学，南京，210023，中国[d]南京师范大学物理科学与技术学院，南京，210023，中国[e]南京大学固体微结构国家实验室，南京，210093，中国

arXiv:2302.07411v1 [cs.CR] 15 Feb 2023

摘要

由于相邻像素之间存在很强的相关性，大多数图像加密方案通过执行多轮混淆与扩散来保护图像免受攻击。然而，此类操作耗时较长，无法满足视频加密的实时要求。因此，现有研究通常通过简化加密过程或仅加密视频帧的特定部分来实现视频加密，这导致其安全性低于图像加密。为解决这一问题，本文提出了一种基于多线程并行混淆与扩散的实时混沌视频加密策略。该策略以视频为输入，将帧划分为子帧，创建一组线程，同时对相应子帧执行五轮混淆与扩散操作，并高效输出加密帧。加密速度评估表明，我们的方法显著提高了混淆与扩散速度，使用 Intel Core i5-1135G7、Intel Core i7-8700 和 Intel Xeon Gold 6226R，分别实现了实时 480×480、 576×576和768×768 24帧率的视频加密。统计分析和安全性分析证明，所部署的密码系统具有出色的统计特性，能够抵抗攻击、信道噪声和数据丢失。据我们所知，与此前的研究相比，所提策略实现了最快的加密速度，并首次实现了基于多轮混淆-扩散架构的实时混沌视频加密，从而为实际应用和相关研究提供了更加安全、可行的解决方案。

*K*关键词： 实时视频加密、并行计算、混沌系统、混淆与扩散 on

1. 引言

随着信息与通信技术的快速发展，图像和视频在数据存储与网络传输方面展现出巨大潜力，因此对图像和视频加密产生了广泛的应用需求 [1]。然而，大多数传统密码方案，如数据加密标准、高级加密标准、RSA 等，都是为保护文本信息而设计的，并不适用于图像和视频 [2]。因此，近年来提出了许多基于不同技术的图像加密协议，其中基于混沌的方法因混沌系统固有的特性而备受关注，这些特性包括遍历性、非周期性、非收敛性，以及对初始条件和控制参数的敏感性等 [7]。大多数基于混沌的图像加密算法包含混淆和扩散阶段 [8]。在前一阶段，像素位置会在整幅图像中进行置乱，而不改变像素值 [9]。在后一阶段，则利用混沌系统生成的字节序列，依次修改像素值 [10]。

这类基于混淆-扩散架构的图像加密协议需要反复执行这两个阶段，直至达到令人满意的安全级别 [11]。显然，这非常耗时，无法满足视频加密的实时要求。因此，现有研究通过简化加密过程或加密视频中的特定像素来实现视频加密[12, 13]。第一类（也称为全加密）方法的例子包括：文献 [14] 选取三种混沌映射生成字节序列，并直接对像素与生成的字节执行异或运算来加密视频；文献 [15]使用生成的字节加密帧，并将加密像素作为反馈，以提高所部署密码系统对明文的敏感性；文献 [16] 对视频帧执行一轮混淆操作，随后对像素与生成的字节执行异或运算，从而实现帧加密。由于实际应用对安全性提出了更高要求，近期发表的大多数研究均基于单轮混淆-扩散架构 [17, 18, 19, 20, 21, 22]。第二类方法也称为选择性加密，属于这一类的算法会加密视频帧中的特定像素，以降低计算复杂度 [23]。显然，这些策略以牺牲安全性为代价实现了高效率。

*通讯作者。电子邮箱： jiangd@nju.edu.cn (Dong Jiang),liangliianglu@nju.edu.cn (Liang-liang Lu)

因此，如何在不降低安全性的前提下实现实时视频加密，已成为亟待解决的问题。然而，在全视频加密领域，相关研究寥寥无几。据我们所知，现有研究无法实现实时视频加密，即每秒加密的帧数大于视频的帧率（每秒帧数），或平均加密时间（毫秒）小于 1000 / FPS。因此，本文利用并行计算，设计了一种基于五轮混淆-扩散架构的实时混沌视频加密策略。为评估性能，我们采用两种不同的混沌映射实现了两个密码系统，并使用三个硬件平台评估所部署密码系统的加密速度。评估结果表明，我们的策略显著提升了字节生成、混淆和扩散的速度，为实现实时视频加密奠定了基础。统计分析和安全性分析证明，所部署的密码系统具有出色的统计特性，并能够抵御攻击、信道噪声和数据丢失。所提策略也适用于多种混淆和扩散方法，且可通过软件和硬件轻松实现。

本文其余部分安排如下：第 2 节详细介绍所提策略。第 3 节选取两种典型的混沌映射来实现该策略，并使用三种不同的硬件平台评估所部署密码系统的加密速度。第 4 节和第 5 节分别进行统计分析和安全性分析。第 6 节分析本文所采用参数设置的原因。第 7 节将本文方法与近期发表的研究工作进行比较，随后在第 8 节给出简要结论。

2. 策略描述

在所提策略中，如图1所示，同时使用一组线程对视频帧进行加密。这些线程可分为两类：一个主线程T_m和 n 个辅助线程T_a^i（$i \in \{0, \ldots, n-1\}$）。主线程以密钥K和原始视频V 作为输入，初始化其伪随机比特生成器m，生成参数P_m，创建并组织T_a线程，对视频帧执行混淆和扩散操作。有关T_m的详细加密过程，请参见算法*1* 。

算法 1： 主线程 T_m 的加密过程。

输入： 密钥：K；辅助线程数量：n；视频文件：V；混淆与扩散操作：r。

输出： 用于初始化T_a的 PRBG$_a$：P_m；明文帧：F_P；混淆种子：s_c。

使用K初始化 PRBG$_m$ ；
迭代 PRBG$_m$，生成足够的迭代结果；
将迭代结果转换为参数P_m；
对于 $i = 0;\ i < n;\ i++$ **执行**
  创建辅助线程 T_a^i;
当从 V **执行**获取一帧 F_P
  迭代伪随机比特生成器m，并生成置乱种子 s_c;
  对于 $i = 0;\ i < r;\ i++$ **执行**
    唤醒 T_as 以执行混淆操作;
    等待 T_as 完成混淆操作;
    唤醒T_a秒以执行扩散操作;
    等待T_a秒以完成扩散操作;

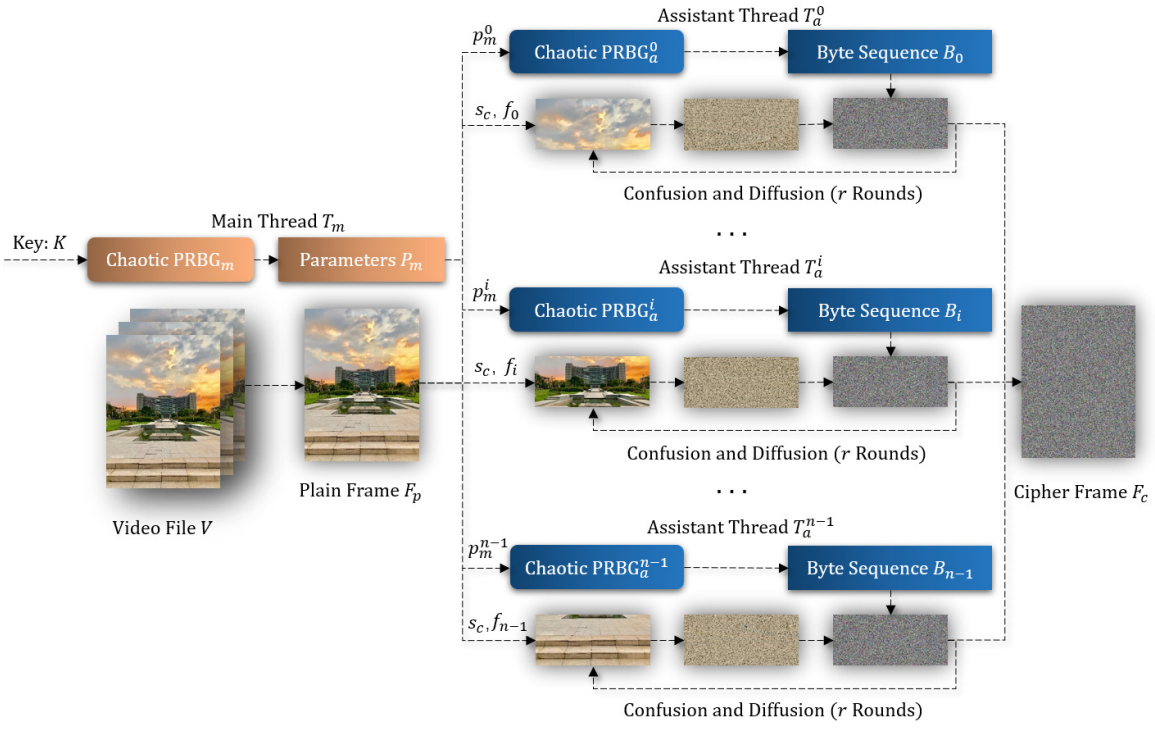

图1：所提混沌实时视频加密策略的工作流程图（p_m：由伪随机比特生成器m生成的参数；s_c：置乱种子；f：子帧）。

辅助线程T_a^i使用参数$p_m^i \in P_m$初始化其伪随机比特生成器，生成迭代结果，将结果转换为字节序列B_i，对子帧f_i执行混淆和扩散操作，并输出密文帧F_c。在混淆操作中，使用如下所示的离散化 Chirikov标准映射 [24] 来重新排列像素位置：

$$\begin{cases} \alpha = (a + o) \bmod w, \\ \beta = (o + s_c \sin(2\pi\alpha/w)) \bmod w, \end{cases} \tag{1}$$

其中，a 和o 分别是像素的横坐标和纵坐标，α 和 β是像素经过混淆操作后的新坐标，w 是视频帧的宽度，而s_c 称为混淆种子，是由T_m生成的一个正整数。混淆操作的逆变换如下：

$$\begin{cases} a = (\alpha - \beta + s_c \sin(2\pi\alpha/w)) \bmod w, \\ o = (\beta - s_c \sin(2\pi\alpha/w)) \bmod w. \end{cases} \tag{2}$$

算法2： 辅助线程 T_a^i 的加密过程。

输入： 线程索引：i；辅助线程数量：n；明文帧：F_P；F_P 的宽度：w；由 T_m 生成的参数：P_m；混淆与扩散轮数r；置乱种子：s_c。

输出： 密文帧：F_c。

起始行 $= i \times w/n$;
结束行 =起始行 $+w/n$;
初始化伪随机比特生成器$_a^i$，使用 p_m^i;
当 从 F_P 中获取子帧 f_i **时**
  迭代伪随机比特生成器$_a^i$，生成足够的迭代结果 ;
  将迭代结果转换为字节序列 B_i;
  $F_c = F_P$;
  对于$j = 0$; $j < r$; $j++$**执行**
    等待被T_m唤醒;
    临时帧$F_t = F_c$;
    对于a = startRow; $a <$ endRow; $a++$**执行**
      对于$o = 0$; $o < w$; $o++$**执行**
        $\alpha = (a + o)$ 模 w;
        $\beta = (o + s_c \sin(2\pi\alpha/w)) \bmod w$;
        $F_c[\alpha, \beta] = F_t[a, o]$;
    告知 T_m 混淆操作已完成
    等待 T_m 唤醒
    $F_t = F_c$;
    取最后一个像素作为扩散种子$f_{(i+1) \bmod n}$ s_d;
    对 a = startRow; $a <$ endRow; $a++$**执行**
      对 $o = 0$; $o < w$; $o++$**执行**
        获取一个字节b，来自B_i;
        如果 a == startRow AND o == 0 **则**
          $F_c[a,o] = b \oplus (F_t[a,o] + b) \oplus s_d$;
        else
          获取前一个像素 $F_c[a',o']$;
          $F_c[a, o] = b \oplus (F_t[a, o] + b) \oplus F_c[a', o']$;
    告知 T_m 扩散操作已完成

在扩散过程中，使用以下方程[25]:

$$F_c[a, o] = b \oplus (F_p[a, o] + b) \oplus F_c[a', o'], \tag{3}$$

其中，$F_c[a,o]$ 和 $F_p[a,o]$ 分别表示加密像素和明文像素，b 是由伪随机比特生成器生成的一个字节$_a^i$，而 $[a',o']$ 是前一个像素的坐标。要执行扩散操作，需要设置扩散种子 s_d 以加密$F_p[0,0]$。在所提出的策略中，T_a^i 取子帧$f_{(i+1) \bmod n}$ 的最后一个像素作为s_d。扩散的逆变换如下：

$$F_p[a, o] = (b \oplus F_c[a, o] \oplus F_c[a', o']) - b. \tag{4}$$

详细的加密过程请参见算法 2，其中 T_a^i。

3. 加密速度评估

为评估所提出策略的加密速度，我们选取了两种引用率很高的混沌映射来实现伪随机比特生成器。第一个伪随机比特生成器使用两个分段线性混沌映射（PLCM） [26]，其定义如下：

$$x_{i+1} = F(x_i, p) = \begin{cases} x_i/p, & x_i \in [0, p) \\ (x_i - p)/(\frac{1}{2} - p), & x_i \in [p, 0.5], \\ F(1 - x_i, p), & x_i \in (0.5, 1] \end{cases} \tag{5}$$

其中，$x_i \in [0,1]$、x_0 和$p \in (0,0.5)$分别表示初始条件和控制参数。第二个伪随机比特生成器采用耦合的二维 Logistic-Adjusted-Sine 映射（2DLASM） [27]，其定义如下：

$$\begin{cases} x_{i+1} = \sin(\pi\mu(y_i + 3)x_i(1 - x_i)), \\ y_{i+1} = \sin(\pi\mu(x_{i+1} + 3)y_i(1 - y_i)), \end{cases} \tag{6}$$

其中，x_i、$y_i \in [0,1]$、x_0 和 $\mu \in [0.37,0.38] \cup [0.4,0.42] \cup [0.44,0.93] \cup \{1\}$ 分别表示初始条件和控制参数。

选取了三个硬件平台进行评估。平台的规格列于表 1。软件开发环境如下：Ubuntu 20.04 操作系统、OpenCV 4.2.0 和 g++ 9.4.0。

表1：硬件平台的规格

平台	规格	
笔记本电脑	CPU	Intel Core i5-1135G7
	核心数 / 线程数	4 / 8
	频率	2.4GHz
	内存	8GB
个人电脑	CPU	Intel Core i7-8700
	核心数 / 线程数	6 / 12
	频率	3.2GHz
	内存	32GB
工作站	CPU	Intel Xeon Gold 6226R
	核心数 / 线程数	16 / 32
	频率	2.9GHz
	内存	64GB

由于所提策略的加密过程由字节生成、混淆和扩散组成，我们使用所选的混沌映射实现了两个密码系统，分别评估这些步骤的速度，并评估已部署密码系统的视频加密速度。在这些实验中，用于初始化伪随机比特生成器的密钥 K 是随机选取的，原始视频从视频跟踪库下载，并转换为不同尺寸以评估加密速度。

首先，同时使用不同数量的辅助线程生成字节序列。在基于 PLCM 的密码系统中，辅助线程 T_a^i 使用主线程 T_m 生成的 p_m^i 初始化其伪随机比特生成器$_a^i$，迭代 PLCM，并生成两组迭代结果 $\{x_1^1, x_2^1, x_3^1...\}, \{x_1^2, x_2^2, x_3^2...\}$；从每个迭代结果的尾数部分[1] 提取 6 个字节，并生成两组字节序列 $\{b_1^1, b_2^1, b_3^1, ...\}, \{b_1^2, b_2^2, b_3^2, ...\}$。通过对生成的字节依次执行异或运算，$T_a^i$ 输出最终字节序列 B_i，可用于视频加密和解密。在基于 2DLASM 的密码系统中，同样地，T_a^i 使用 p_m^i 初始化其伪随机比特生成器$_a^i$，迭代 2DLASM，并生成两组迭代结果$\{x_1^1, y_1^1, x_2^1, y_2^1, ...\}, \{x_1^2, y_1^2, x_2^2, y_2^2, ...\}$。通过从每个迭代结果中提取 6 个字节，并对生成的字节执行异或运算，T_a^i 生成字节序列 B_i。为计算吞吐量，所有混沌映射均迭代 5×10^7 次。吞吐量（MBps）与辅助线程数量之间的关系如图 2 (a)-(c) 所示。

其次，我们将混淆轮数和扩散轮数设为5（原因见第 6 节），使用不同数量的辅助线程对 100 张尺寸为 960×960 的图像执行五轮混淆操作，计算图像混淆的平均耗时，并将结果绘制在图 2 (d)-(f) 中。同样，我们使用不同数量的辅助线程生成字节序列，利用生成的字节对 100 张尺寸为 960×960 的图像执行五轮扩散操作，并计算完成一张图像五轮扩散操作的平均耗时。平均扩散耗时（包括字节生成和扩散操作）与辅助线程数量之间的关系如图 2 (g)-(i) 所示。与字节生成、混淆和扩散的单线程版本相比，显然，所提策略显著加快了这些步骤。

第三，为评估所提策略的实际性能，使用已部署的密码系统对一组不同尺寸的视频进行加密。与上述实验类似，混淆和扩散轮数ff设为 5。辅助线程数ff r 设为 8、12 和 32，分别对应笔记本电脑、个人电脑和工作站（原因见第 6 节）。所选视频的帧率为 20，即加密一帧所需的最长时间必须小于 50 毫秒，否则加密和传输过程中会出现延迟。每个视频包含 600 帧。平均加密时间列于表 2，结果表明，使用 Intel Core i5-1135G7、Intel Core i7-8700 和 Intel Xeon Gold 6226R 分别实现了实时视频加密 576×576，672×672，以及 960×960 视频加密。

[1]在 IEEE 754 标准中，双精度浮点数由 1 位符号位、11 位指数和 52 位尾数组成。

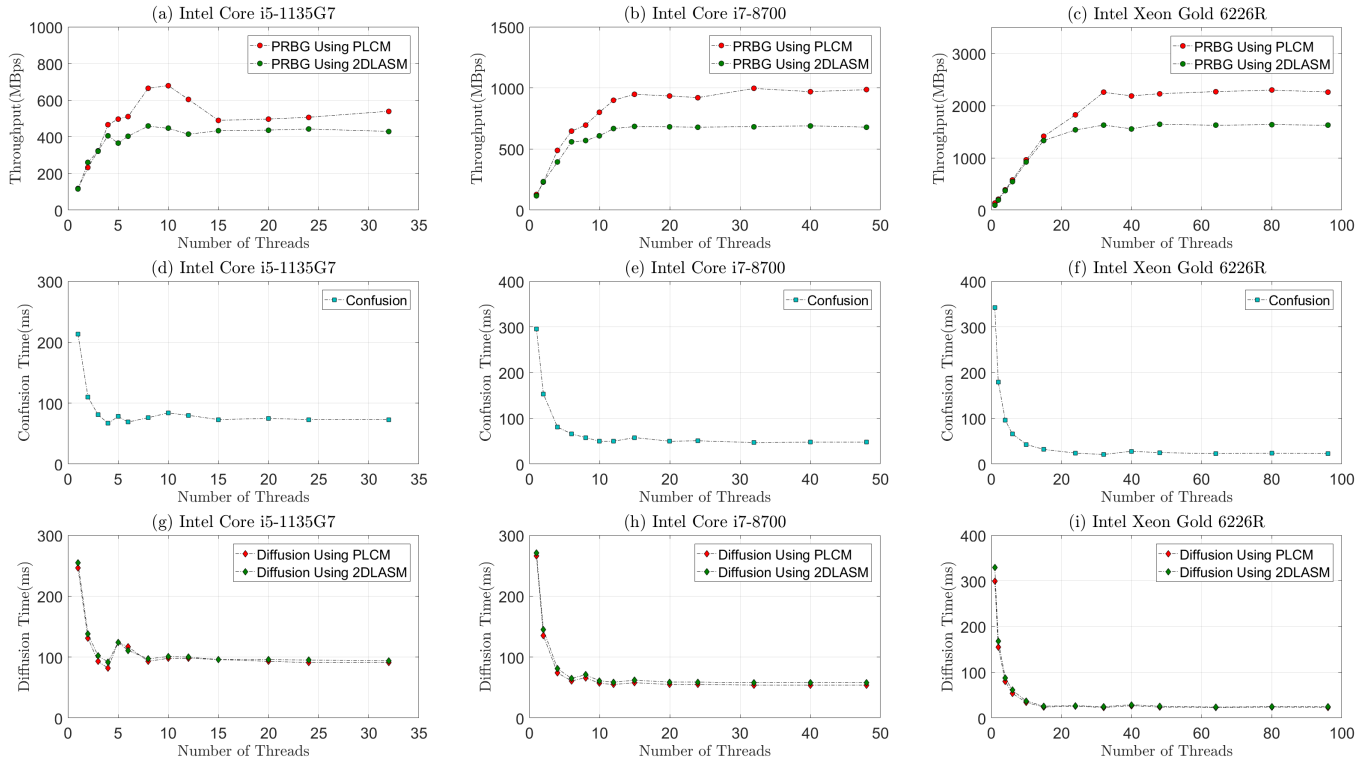

图 2：字节生成、混淆和扩散步骤的速度评估；(a-c) 所有伪随机比特生成器$_a$的吞吐量与辅助线程数量的关系；(d-f) 混淆操作的平均时间与辅助线程数量的关系；(g-i) 扩散操作的平均时间与辅助线程数量的关系。

表 2：视频加密速度评估。

视频大小	最大加密时间	笔记本电脑（8T_as）平均加密时间（毫秒）		个人电脑（12 T_as）平均加密时间（毫秒）		工作站（32 T_as）平均加密时间（毫秒）	
		PLCM	2DLASM	PLCM	2DLASM	PLCM	2DLASM
96×96	42ms	5.69	5.54	7.43	7.40	2.92	2.62
192×192	42ms	16.59	16.56	21.45	20.92	7.69	7.41
288×288	42ms	29.63	28.98	30.39	31.76	15.12	14.94
384×384	42ms	31.60	30.64	35.97	35.22	23.25	22.94
480×480	42ms	36.26	35.49	37.76	37.88	28.76	28.11
576×576	42ms	×	×	39.92	40.67	30.40	30.02
672×672	42ms	×	×	×	×	33.60	32.89
768×768	42ms	×	×	×	×	36.56	36.23

帧数：300；帧率：24；混淆与扩散轮数：$r = 5$；×：平均加密时间 ≥ 最大加密时间 .

4. 统计评估

实时视频加密的本质是图像加密，它应具备优异的统计特性，以抵御不同类型的攻击ff。因此，本节将分析已部署密码系统的均匀性、相关性和随机性。以下实验使用笔记本电脑（Intel Core i5-1135G7）进行，参数设置如下：辅助线程数：$n = 8$，混淆和扩散轮数ff $r = 5$，用于统计评估的所有明文图像尺寸为 512×512。

4.1. 均匀性评估

直方图、方差和 χ^2 用于评估所部署密码系统的均匀性。明文图像如图 3 (a) 所示，其红色、绿色和蓝色通道的直方图分别如图 3 (b)、(c)、(d) 所示。使用 PLCM 加密的密文图像如图 3(e) 所示。其红色、绿色和蓝色通道的直方图分别如图3 (f)、(g)、(h) 所示。使用 2DLASM 加密的密文图像如图 3 (f) 所示。其红色、绿色和蓝色通道的直方图分别如图 3 (j)、(k)、(l) 所示。

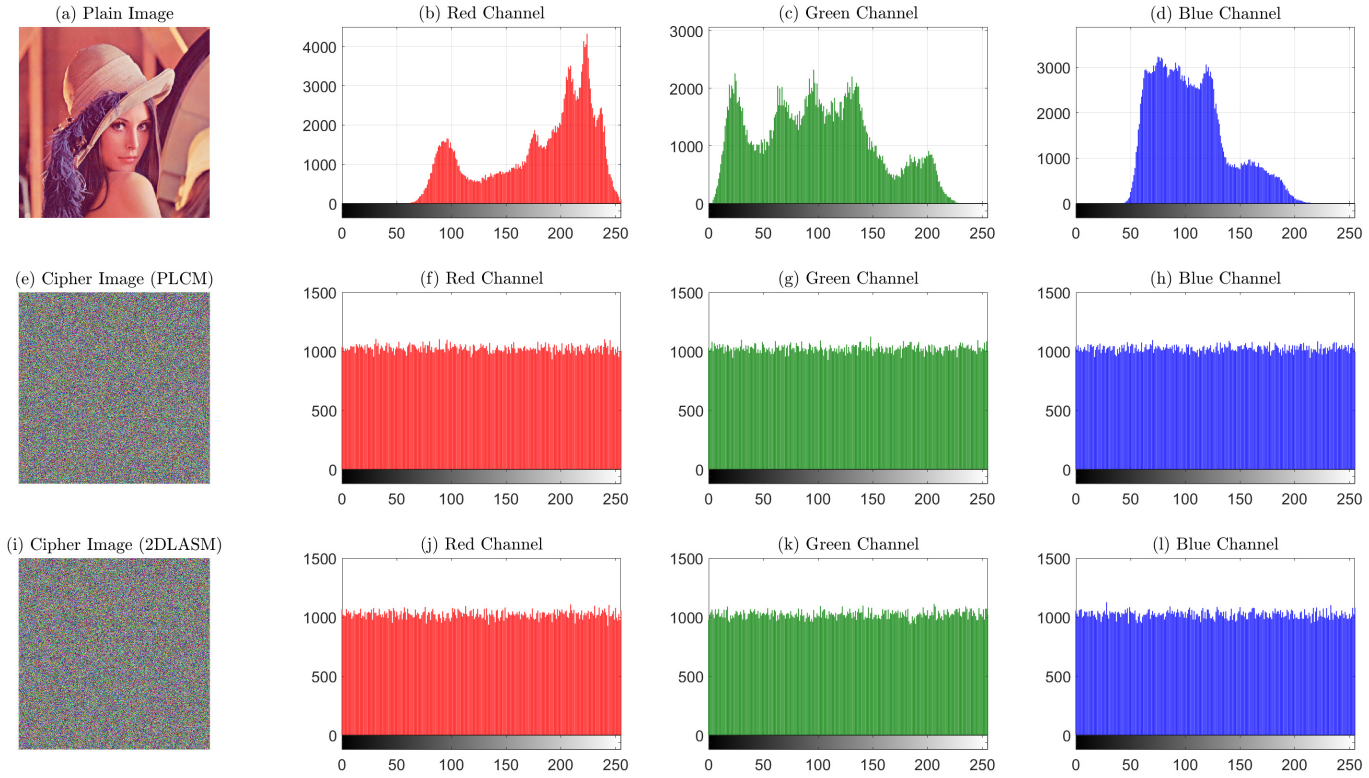

图3：明文图像和密文图像的直方图。（a）明文图像 Lena；（b）–（d）明文图像的红色、绿色和蓝色通道的直方图；（e）使用 PLCM 加密的密文图像；（f）–（h）使用 PLCM 加密的密文图像的红色、绿色和蓝色通道的直方图；（i）使用 2DLASM 加密的密文图像；（h）–（j）使用 2DLASM 加密的密文图像的红色、绿色和蓝色通道的直方图。

图像的方差可按如下方式计算 [29]:

$$\mathrm{var(Z)} = \frac{1}{256^2}\sum_{i=0}^{255}\sum_{j=0}^{255}\frac{1}{2}(z_i - z_j)^2, \tag{7}$$

其中，$\mathrm{Z} = \{z_0, z_1, ..., z_{255}\}$ 是直方图值向量，z_i 和 z_j 分别是取值等于 i和 j 的像素数量。图像的 χ^2 由 [30]给出:

$$\chi^2 = \sum_{i=0}^{255}\frac{(z_i - N/256)^2}{N/256}, \tag{8}$$

表3：明文图像和密文图像的方差及 χ^2 结果

Test	图像	通道		
		R	G	B
var	Lena	1021324	457505	1382757
	Lena (PLCM)	920.81	982.86	802.76
	Lena (2DLASM)	1078.06	1047.92	1065.47
	飞机	2724339	2740687	4448810
	飞机（PLCM）	1139.74	1043.63	930.89
	飞机（2DLASM）	1022.57	1115.45	1035.92
	Lake	789874	522660	1383691
	湖泊（PLCM）	942.18	1019.14	1166.58
	湖泊（2DLASM）	1063.56	913.51	1068.22
	辣椒	856092	1278525	1973421
	辣椒 (PLCM)	981.84	995.96	919.01
	辣椒 (2DLASM)	1030.07	946.35	1101.70
	山魈	332658	573472	321024
	山魈 (PLCM)	1018.60	1016.43	1138.86
	山魈 (2DLASM)	1165.99	993.28	872.41
	水花	2432151	3095990	5940168
	水花 (PLCM)	1015.49	1112.96	1072.76
	水花 (2DLASM)	954.40	880.06	1127.92
χ^2	Lena	254333	113929	344338
	Lena (PLCM)	229.30	244.75	199.90
	Lena (2DLASM)	268.46	260.95	265.32
	飞机	678424	682495	1107858
	飞机 (PLCM)	283.82	259.88	231.81
	飞机 (2DLASM)	254.64	277.77	257.96
	Lake	196697	130154	344571
	湖泊 (PLCM)	234.62	253.79	290.50
	湖泊 (2DLASM)	264.85	227.48	266.01
	辣椒	213187	318382	491428
	辣椒 (PLCM)	244.50	248.01	228.85
	辣椒 (2DLASM)	256.51	235.66	274.34
	山魈	82839	142808	79942
	山魈（PLCM）	253.65	253.11	283.60
	山魈（2DLASM）	290.35	247.35	217.25
	水花	605662	770974	1479241
	水花（PLCM）	252.88	277.15	267.14
	Spalsh（2DLASM）	237.66	219.15	280.88

其中 N 为像素数量，$N/256$ 为预期出现频率。方差越高，表示图像的均匀性越低；当显著性水平为 0.05 时，$\chi^2(0.05,255)$ 等于 293。25 [31]。我们分别使用 PLCM 和 2DLASM 对一组图像进行加密，计算并列出了明文图像和密文图像的方差以及 χ^2 测试结果，见表 3。显然，与对应的明文图像相比，密文图像的方差显著降低，且 χ^2 密文图像的数值低于 293.25。因此，所采用的密码系统能够均匀地隐藏明文图像的信息。

4.2. 相关性评估

由于图像中的两个相邻像素通常具有较高的相关性，密码系统应降低这种相关性，以抵抗统计攻击 [32]。因此，我们分析了明文图像和密文图像中两个相邻像素之间的相关性。首先，我们从明文图像和密文图像中随机选取 6000 对水平、垂直和对角相邻的像素。明文图像中红色、绿色和蓝色通道的相关性分布分别如图 4(a)、(b) 和 (c) 所示。密文图像中红色、绿色和蓝色通道的相关性分布分别绘制于图 4 (d)、(e) 和 (f) 中。密文图像中红色、绿色和蓝色通道的相关性分布如图 3 (i) 所示，分别绘制于图 4 (g)、(h) 和 (i) 中。

然后，测量相邻像素之间的相关系数。相邻像素的相关系数$r_{x,y}$ 可根据以下公式:计算得出:

$$r_{x,y} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)D(y)}}, \tag{9}$$

其中，x 和y 是图像中两个相邻像素的值，cov(x,y)的计算公式为

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{10}$$

N 是从图像中选取用于计算的像素总数，$E(x)$和$D(x)$定义如下:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i, \tag{11}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2. \tag{12}$$

密文图像的相关系数越低，加密算法抵抗统计分析攻击的能力就越强。为了评估所提策略的相关性，我们使用 PLCM 和 2DLASM 对一组明文图像进行加密，从明文图像和密文图像中随机选取 20000 对水平方向、垂直方向和对角方向上的相邻像素，计算相关系数，并将结果列于表 4。显然，所部署的密码系统显著降低了明文图像的相关性。

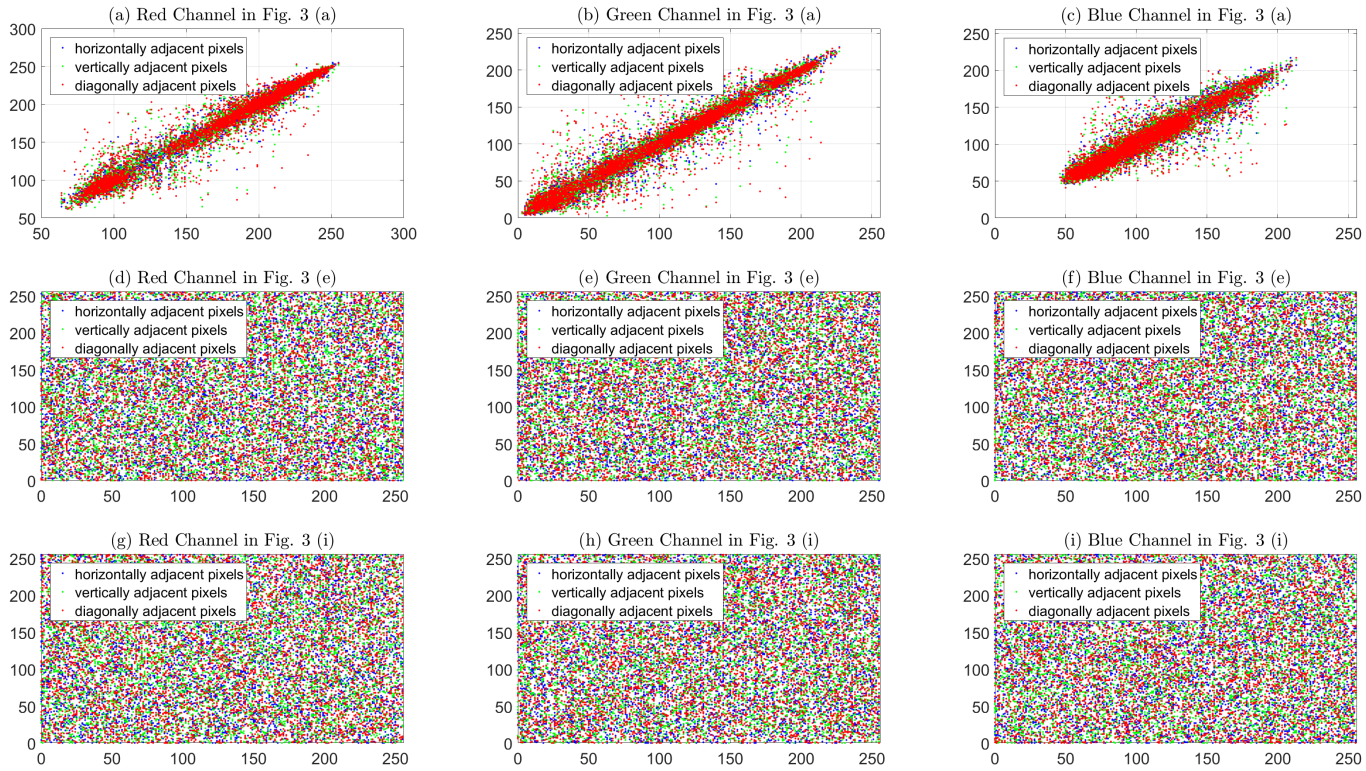

图 4：水平、垂直和对角方向上两个相邻像素的相关性分布；(a) - (c) 明文图像中红色、绿色和蓝色通道的相关性分布；(d) - (f) 使用 PLCM 加密的密文图像中红色、绿色和蓝色通道的相关性分布；(g) - (i) 使用 2DLASM 加密的密文图像中红色、绿色和蓝色通道的相关性分布。

表 4：明文图像和密文图像中两个相邻像素的相关系数

图像	通道	明文图像			使用 PLCM 加密的密文图像			使用 2DLASM 加密的密文图像		
		H	V	D	H	V	D	H	V	D
Lena	R	0.988469	0.980273	0.968900	-0.003864	-0.000723	-0.009886	-0.001331	-0.003863	-0.002300
	G	0.980549	0.968197	0.951883	0.013827	0.003009	0.002088	-0.008544	-0.002600	-0.012616
	B	0.951984	0.927028	0.908143	0.003075	0.002843	0.003700	-0.002474	0.001525	-0.016043
飞机	R	0.970642	0.970695	0.945469	0.006467	-0.006111	0.003168	0.001575	0.011213	-0.008233
	G	0.971925	0.926737	0.902899	-0.004944	0.003830	-0.017420	-0.003714	0.001359	-0.008748
	B	0.956298	0.959205	0.930419	-0.001179	-0.000550	-0.002195	-0.001821	-0.000839	0.003026
Lake	R	0.954300	0.950933	0.938652	-0.002712	-0.002231	-0.007473	0.008294	0.002791	0.005007
	G	0.968942	0.972935	0.957351	-0.006042	0.005319	-0.003452	-0.005502	-0.003010	-0.000647
	B	0.968070	0.969546	0.951428	0.005635	-0.003997	0.002186	0.010300	-0.003263	0.001231
辣椒	R	0.968357	0.956775	0.950391	-0.000807	-0.004586	-0.001847	-0.002638	0.007745	0.005794
	G	0.986690	0.972400	0.964351	0.004994	0.011397	0.008190	0.003480	0.004427	0.004097
	B	0.972810	0.959227	0.947239	-0.003806	0.007530	-0.006979	-0.003785	-0.004886	-0.001596
山魈	R	0.853960	0.910030	0.841495	0.000305	-0.009033	0.004641	0.010308	0.004073	0.007677
	G	0.771212	0.858922	0.739454	0.000354	0.006221	-0.000766	-0.008846	-0.006407	0.000659
	B	0.864290	0.892019	0.825914	0.005260	-0.007633	0.001064	0.002351	-0.010925	0.009007
水花	R	0.985901	0.992967	0.980009	0.012251	-0.003002	-0.005989	-0.002299	0.005270	0.005558
	G	0.979680	0.981868	0.965632	0.006494	0.003150	-0.006994	0.001839	-0.002984	0.001348
	B	0.959442	0.983775	0.948882	0.015445	0.003662	-0.009827	-0.006020	-0.010502	0.002206

H：水平相关系数；V：垂直相关系数；D：对角相关系数。

4.3. 随机性评估

本节采用三种广泛使用的方法，即信息熵、(k,T_B) 局部香农熵和 NIST统计测试，来评估所提策略的随机性属性。信息熵的定义用于量化信息的随机性 [34]。对于给定的信息m，熵 $H(m)$ 可按如下方式计算：

$$H(m) = \sum_{i=1}^{N} p(m_i)\log\frac{1}{p(m_i)}, \tag{13}$$

表 5：普通图像和密文图像的信息熵 H 及 (k,T_B)-局部香农熵 H_{k,T_B}。

Test	图像	通道		
		R	G	B
H	Lena	7.253102	7.594038	6.968427
	Lena (PLCM)	7.999369	7.999326	7.999449
	Lena（2DLASM）	7.999259	7.999282	7.999269
	飞机	6.717765	6.798979	6.213774
	飞机（PLCM）	7.999219	7.999286	7.999361
	飞机（2DLASM）	7.999299	7.999234	7.999291
	Lake	7.312388	7.642854	7.213642
	湖泊（PLCM）	7.999353	7.999302	7.999199
	湖泊（2DLASM）	7.999271	7.999373	7.999268
	辣椒	7.338827	7.496253	7.058306
	辣椒 (PLCM)	7.999328	7.999317	7.999370
	辣椒 (2DLASM)	7.999292	7.999353	7.999245
	山魈	7.706672	7.474432	7.752217
	山魈 (PLCM)	7.999302	7.999302	7.999221
	山魈 (2DLASM)	7.999201	7.999319	7.999403
	水花	6.948061	6.884455	6.126452
	水花（PLCM）	7.999305	7.999238	7.999263
	水花（2DLASM）	7.999346	7.999397	7.999225
$\overline{H}_{k,\ T_B}$	Lena	5.911418	6.378169	6.044504
	Lena（PLCM）	7.901188	7.900454	7.902854
	Lena（2DLASM）	7.903580	7.902076	7.901568
	飞机	5.371063	5.348903	4.918050
	飞机（PLCM）	7.903643	7.901751	7.902929
	飞机（2DLASM）	7.903149	7.904907	7.901914
	Lake	6.129053	6.348979	5.847770
	湖泊（PLCM）	7.906276	7.901353	7.901770
	湖泊（2DLASM）	7.902562	7.900261	7.901848
	辣椒	5.975824	6.057454	5.870517
	辣椒 (PLCM)	7.903389	7.905916	7.901726
	辣椒 (2DLASM)	7.902408	7.899511	7.904691
	山魈	6.578668	6.688874	6.802597
	山魈 (PLCM)	7.903176	7.901764	7.903579
	山魈 (2DLASM)	7.901823	7.901584	7.903498
	水花	4.150002	5.184026	4.560897
	水花 (PLCM)	7.903036	7.902193	7.903928
	Spalsh (2DLASM)	7.900637	7.904737	7.902950

其中，N 是符号总数，$p(m_i)$ 是符号 m_i 的出现概率。对于随机图像 F，熵 $H(F)$ 等于 8。因此，密码系统应生成信息熵接近 8 的密文图像。

采用 (k,T_B) 局部香农熵来衡量图像的局部随机性 [35]，其定义如下：

$$\overline{H}_{k,T_B} = \sum_{i=1}^{k} \frac{H(M_i)}{k}, \tag{14}$$

其中，$M_i (i \in \{1,2,...,k\})$ 是随机选取的互不重叠的图像子块，每个子块包含 T_B 个像素，$H(M_i)$ 是子块 M_i的信息熵。我们使用随机选取的密钥 K 分别通过 PLCM 和 2DLASM 加密一组明文图像，信息熵结果列于表 5。然后根据参考文献 [36, 37], , 将 k 和 T_B 分别设为 30 和 1936。随机从明文图像和生成的密文图像中选取互不重叠的子块，计算 $H_{k、T_B}$，并将结果列于表 5。

NIST统计测试套件（版本800-22）可用于测试随机或伪随机数生成器生成的序列是否适用于密码学应用 [38]。该套件包含15项统计检验，针对比特序列中可能存在的不同类型的非随机性。通过这些测试，可以在显著性水平 α 下判断理想随机性假设是否被接受，该显著性水平设为0。01。对于每项测试，P值和通过率r_P 被视为判定结果。当P值大于显著性水平α时，则认为该序列是随机的。在视频加密中，辅助线程获取一个子帧，生成字节序列，并对该子帧执行扩散操作。根据加密过程，我们使用8个辅助线程$T_a^i\, i \in \{1,2,.....、\ 8\}$生成足够的字节序列$B_1, B_2,.....、B_8$分别使用PLCM和2DLASM。随后对生成的字节执行NIST测试。结果如表6所示。

表 6：NIST测试结果。

统计检验	PLCM		2DLASM	
	P值	r_P(%)	P值	r_P(%)
频率	0.181557	99	0.657933	98
块频率	0.474986	99	0.955835	99
累积和*	0.452541	99	0.717629	98
Runs	0.350485	99	0.115387	97
最长游程	0.883171	97	0.171867	100
Rank	0.366918	99	0.554420	100
FFT	0.897763	100	0.595549	100
非重叠模板*	0.486724	99	0.510463	98
重叠模板	0.366918	98	0.883171	96
通用性	0.455937	100	0.574903	97
近似熵	0.983453	99	0.249284	98
随机游动*	0.415715	99	0.258140	99
随机游动变体*	0.442424	99	0.276585	99
序列检验*	0.428515	99	0.944186	99
线性复杂度	0.137282	96	0.058984	100

显著性水平 $\alpha = 0.01$；r_P：通过率；$*$：平均值。

5. 安全性分析

5.1. 密钥空间与敏感性

密码系统安全性最重要的问题之一是密钥。根据Kerckhoff原则，即使破解者知道系统除密钥之外的一切信息，密码系统仍应保持安全 [39]。如果密钥选择不当或密钥空间过小，那么无论加密算法多么强大，密码系统都很容易被破解。通常认为，密码系统的密钥空间应足够大 2^{100}，以提供针对唯密文攻击的充足安全性 [40]。对于所提策略，显然，密钥就是用于初始化主线程$_m$ 的伪随机比特生成器T_m的参数。在基于PLCM的密码系统中，取四个双精度浮点数作为初始条件和控制参数，以初始化两个PLCM。在基于2DLASM的密码系统中，将三个双精度浮点数分别作为初始条件x_0, 和控制参数μ，以及y_0 输入以初始化2DLASM。因此，基于PLCM和2DLASM的密码系统的密钥空间分别为 2^{256} 和 2^{192}, 。这显然增强了对唯密文攻击的抵抗能力。用户可以采用更复杂的伪随机比特生成器来获得更大的密钥空间，从而提高安全性。

此外，密码系统在加密和解密过程中应对密钥变化具有较高的敏感性。也就是说，即使密钥只改变一位，攻击者也无法从解密图像中获取任何信息。为了分析所提策略的密钥敏感性，我们随机选择密钥，使用部署的密码系统加密图像，然后使用正确的密钥解密密文图像。这些图像使用 PLCM 和 2DLASM 解密得到的结果分别绘于图 5(a) 和 (e) 中。随后，我们在初始条件和控制参数上分别增加 $\delta = 0.000000001$，对密钥稍作修改，并使用修改后的密钥对密文图像进行解密。使用基于 PLCM 的密码系统、分别采用 $x_0+\delta$、$p+\delta$和 $x_0+\delta$、$p+\delta$ 解密得到的图像如图 5 (b)、(c) 和 (d) 所示。使用基于 2DLASM 的密码系统、分别采用 $x_0+\delta$、$\mu+\delta$和 $x_0+\delta$、$\mu+\delta$ 解密得到的图像如图 5 (f)、(g) 和 (h) 所示。我们还计算了使用正确密钥解密的图像与使用稍作修改的密钥解密的图像之间的相关系数，并将结果列于表 7。

表7：图中解密图像之间的相关系数5。

PRBG	图像	通道		
		R	G	B
PLCM	图 5(a),(b)	-0.002588	0.005312	0.002903
	图 5（a）、（c）	-0.001788	0.001664	-0.001074
	图 5(a)、(d)	-0.002994	0.001132	0.002672
	图 5(b),(c)	0.004292	-0.001097	-0.000042
	图5（b）、（d）	0.002267	0.001098	0.002168
	图 5(c),(d)	0.000715	0.004356	0.000954
2DLASM	图 5(e)、(f)	-0.002142	0.000862	0.000712
	图 5(e)、(g)	-0.002620	0.001053	-0.000951
	图 5(e)、(h)	0.003072	-0.000513	0.001539
	图 5(f)、(g)	-0.001342	-0.000709	0.002444
	图 5(f)、(h)	0.004969	-0.001565	0.000370
	图5（g）、（h）	0.003579	0.001962	0.000740

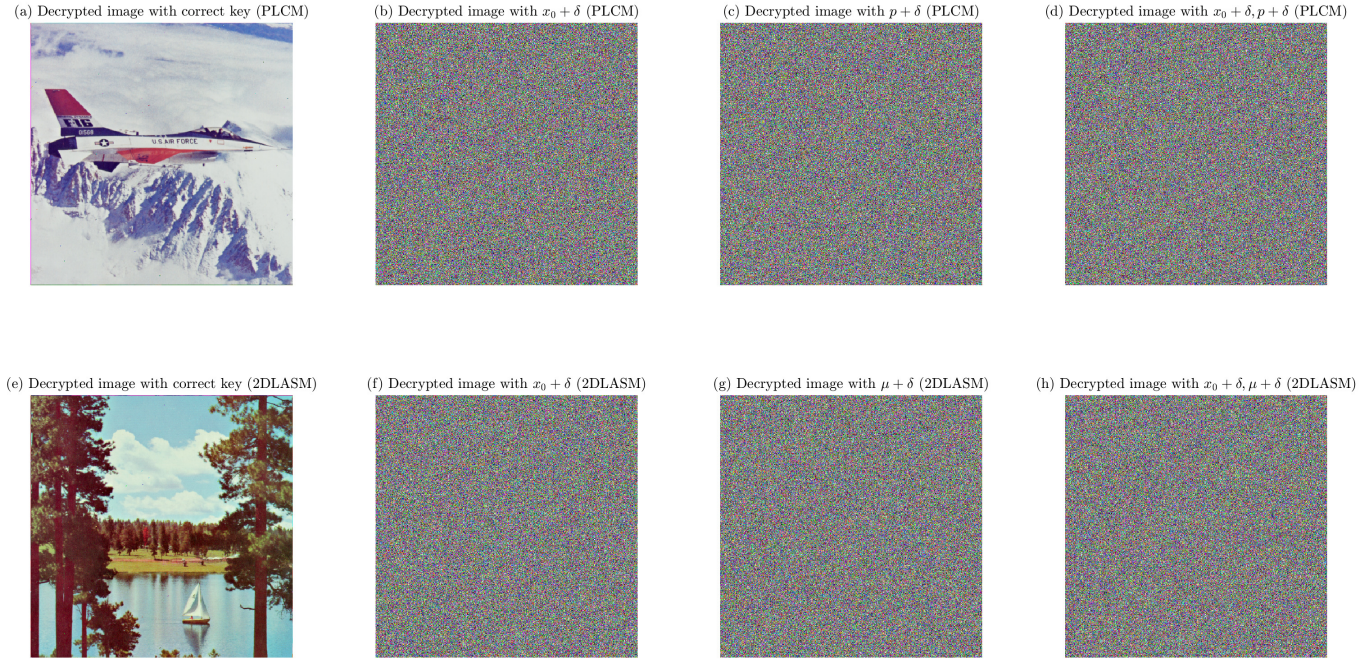

图5：密钥敏感性分析。（a）使用正确密钥通过 PLCM 解密图像，（b）使用 $x_0+\delta$ 通过 PLCM 解密图像，（c）使用$p+\delta$ 通过 PLCM 解密图像，（d）使用 $x_0+\delta$ 和 $p+\delta$ 通过 PLCM 解密图像，（e）使用正确密钥通过 2DLASM 解密图像，（f）使用 $x_0+\delta$ 通过 2DLASM 解密图像，（g）使用 $\mu+\delta$ 通过 2DLASM 解密图像，（h）使用 $x_0+\delta$ 和 $p+\delta$ 通过 2DLASM 解密图像。增量 δ 设为 1×10^{-9}。

5.2. 对差分攻击的抵抗能力

为抵抗差分攻击，加密算法应对明文图像具有较高的敏感性，即明文图像的微小变化会导致密文图像发生彻底变化 [41]。为评估所提出策略抵抗此类攻击的能力，我们计算了像素变化率（NPCR）和统一平均变化强度（UACI） [42]。NPCR 可根据下式计算：

$$\mathrm{NPCR} = \sum_{i=0}^{w}\sum_{j=0}^{h}\frac{D(i,j)}{w\times h}\times 100\%, \tag{15}$$

其中，w 和h 分别表示图像的宽度和高度，$D(i,j)$定义如下：

$$D(i,j) = \begin{cases} 1, & F_c^1[i,j] \neq F_c^2[i,j], \\ 0, & F_c^1[i,j] = F_c^2[i,j], \end{cases} \tag{16}$$

F_c^1和F_c^2分别是使用明文图像和略有变化的明文图像生成的两幅密文图像，$F_c[i,\ j]$ 表示像素值。UACI 的计算公式为：

$$\mathrm{UACI} = \sum_{i=0}^{w}\sum_{j=0}^{h}\frac{|F_c^1[i,j] - F_c^2[i,j]|}{w\times h\times 255}\times 100\%, \tag{17}$$

根据参考文献 [43, 44], , 对于尺寸为 512×512 的图像，当显著性水平等于 0.05 时，像素变化率和 UACI 的期望值分别为 99.5893% 和 [33.3730%，33.5541%]。当像素变化率高于期望值且 UACI 落在该区间内时，图像加密方案通过扩散特性测试。

在所提策略中，尽管所有辅助线程都独立执行扩散操作，但它们会将下一个子帧的最后一个像素作为扩散种子，以重建所有子帧之间的关系。这确保任何子帧中只要有一个像素发生变化，就会产生完全不同的密文帧。为全面评估所提策略抵御差分攻击的能力，我们生成字节序列，对一组明文图像进行加密，随机选取并更改明文图像中的一个像素，再使用相同的字节序列加密修改后的图像，并计算生成的密文图像之间的 NPCR 和 UACI。对于每幅明文图像，我们将上述步骤重复 100 次，获取最小值、最大值和平均值，并在表 8中列出结果。

表 8：像素变化率和 UACI 的结果

Test	图像	PRBG	通道								
			R			G			B		
			Min	Max	平均值	Min	Max	平均值	Min	Max	平均值
NPCR	Lena	CPLCM	99.5770	99.6334	99.6093	99.5762	99.6361	99.6096	99.5781	99.6372	99.6048
		2DLASM	98.9208	99.6376	99.5696	99.1913	99.6395	99.5667	99.2020	99.6414	99.5839
	飞机	CPLCM	99.5800	99.6471	99.6092	99.5789	99.6426	99.6093	99.5861	99.6418	99.6091
		2DLASM	98.4215	99.6349	99.5496	98.4520	99.6433	99.5409	98.4180	99.6403	99.5138
	Lake	CPLCM	98.5302	99.6841	99.5808	96.8250	99.6315	99.5587	98.4116	99.6395	99.5757
		2DLASM	98.6176	99.6284	99.5616	99.1982	99.6384	99.5949	99.1920	99.6445	99.5709
	辣椒	CPLCM	98.4627	99.6361	99.5762	98.4226	99.6510	99.5301	99.2138	99.6407	99.5908
		2DLASM	97.2801	99.6559	99.5536	97.3145	99.6437	99.5582	98.6259	99.6391	99.5693
	山魈	CPLCM	97.2404	99.6464	99.5532	98.8369	99.6407	99.5809	98.4257	99.6304	99.5529
		2DLASM	98.4825	99.7314	99.5763	98.7518	99.6433	99.5658	98.6263	99.6418	99.5753
	水花	CPLCM	98.6454	99.6368	99.5792	98.4016	99.6349	99.5604	98.8396	99.6330	99.5838
		2DLASM	98.4062	99.6349	99.5779	98.4413	99.6376	99.5171	98.4142	99.6357	99.5463
UACI	Lena	CPLCM	33.3000	33.5896	33.4585	33.3756	33.6102	33.4633	33.3486	33.5817	33.4616
		2DLASM	33.3106	33.6019	33.4648	33.3245	33.5847	33.4678	33.3580	33.5815	33.4674
	飞机	CPLCM	33.3524	33.5590	33.4615	33.3716	33.5565	33.4650	33.3379	33.5907	33.4682
		2DLASM	33.3349	33.5936	33.4680	33.3895	33.6084	33.4678	33.3561	33.5746	33.4734
	Lake	CPLCM	33.3271	33.5673	33.4630	33.3637	33.5900	33.4675	33.3601	33.5879	33.4668
		2DLASM	33.3629	33.5763	33.4700	33.3610	33.5877	33.4558	33.3493	33.5729	33.4652
	辣椒	CPLCM	33.3690	33.5836	33.4623	33.3517	33.5588	33.4577	33.3644	33.5995	33.4632
		2DLASM	33.3299	33.5721	33.4585	33.3322	33.5540	33.4664	33.3713	33.5711	33.4687
	山魈	CPLCM	33.3343	33.5772	33.4643	33.3393	33.5597	33.4644	33.3561	33.5829	33.4603
		2DLASM	33.3706	33.5938	33.4720	33.3503	33.5429	33.4555	33.3110	33.6242	33.4626
	水花	CPLCM	33.3785	33.5953	33.4662	33.3411	33.5438	33.4536	33.3535	33.5860	33.4630
		2DLASM	33.3427	33.5976	33.4647	33.3602	33.6073	33.4600	33.3805	33.5635	33.4619

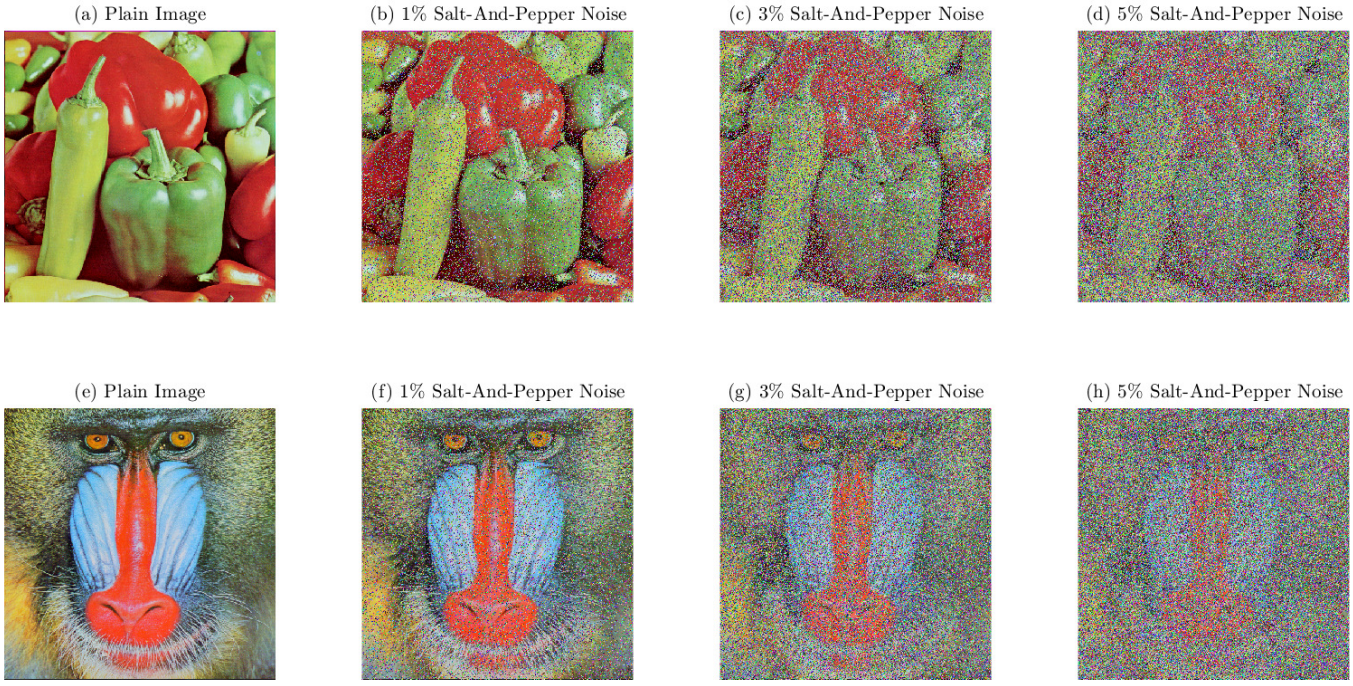

图 6：抗噪性。(a) 明文图像辣椒；(b)–(d) 使用 PLCM 对含有 1%、3%、5% 椒盐噪声的密文图像进行解密；(e) 明文图像山魈；(f)–(h) 使用 2DLASM 对含有 1%、3%、5% 椒盐噪声的密文图像进行解密。

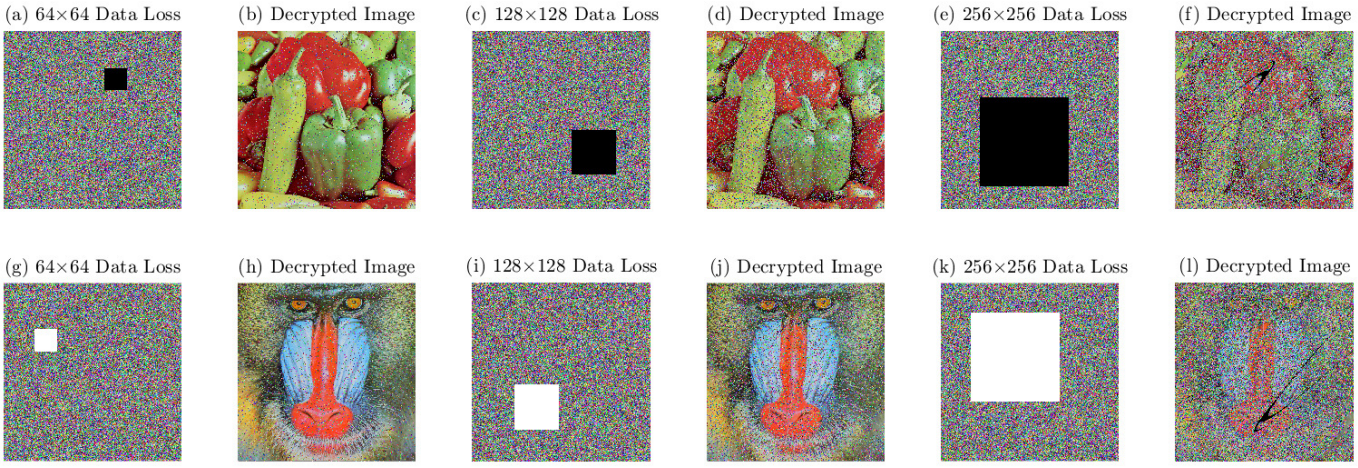

图 7：抗数据丢失性。(a) –(f) 具有不同数据丢失程度的密文图像及使用 PLCM 解密得到的相应图像；(g)–(j) 具有不同的数据丢失及其对应的彩在线 使用 2DLASM 解密。

5.3. 抗噪性和抗数据丢失性

图像或视频帧在通道中传输时可能会受到噪声影响。当密文图像出现噪声或部分数据丢失时，密码系统需要以较高的视觉质量恢复原 [45]始图像。为评估鲁棒性，我们使用两种已部署的密码系统，采用随机选取的密钥加密一幅明文图像，向生成的密文图像 [46] 中添加 1%、3% 和 5% 的椒盐噪声，再使用相同的密钥解密密文图像，并在图 6中绘制明文图像和解密图像。尽管密文图像存在噪声时，解密图像会发生变化，但明文图像的大致信息仍得以保留，即使椒盐噪声高达 5% 也是如此。

在实际应用中，图像裁剪非常常见，这可能导致数据丢失 [47]。为了评估所提策略抵抗数据丢失的能力，同样地，我们使用已部署的密码系统加密一幅明文图像，随机选取并移除 64×64，128×128，或 512×512 个块，并用白色或黑色块替换这些移除的块，再使用相同的密钥解密修改后的密文图像，结果如图 7所示。显然，即使密文图像中有 256×256 个块被移除，明文图像的轮廓仍得以保留。也就是说，当密文图像中存在 25% 的数据丢失时，已部署的密码系统仍能恢复明文图像的信息。因此，所提策略能够抵抗噪声和数据丢失。

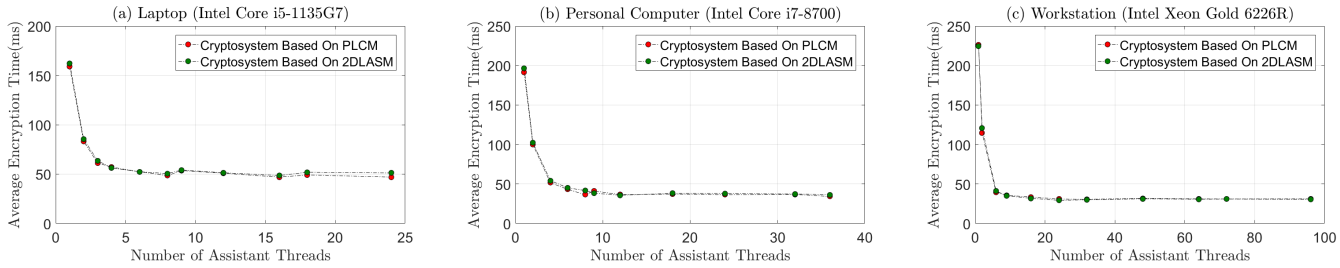

图8：辅助线程数量与加密速度之间的关系：(a) 笔记本电脑（Intel Core i5-1135G7）的结果，(b) 个人电脑（Intel Core i7-8700G7）的结果，(c) 工作站（Intel Xeon Gold 6226R）的结果。

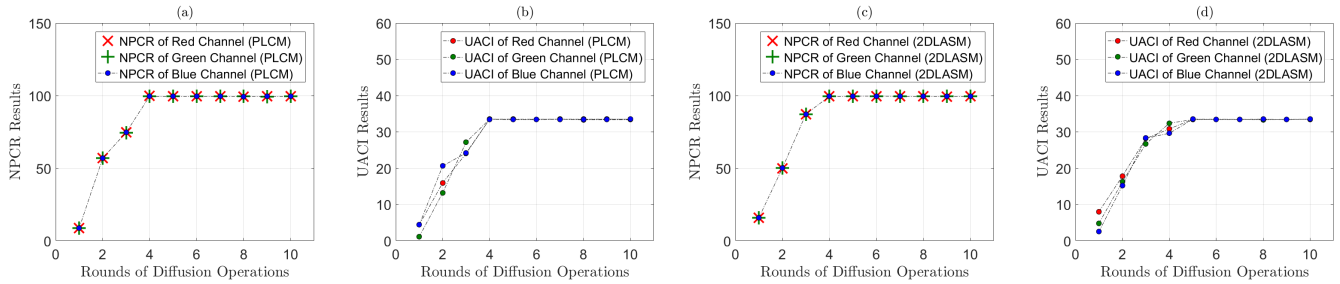

图9：扩散操作轮数与像素变化率、UACI结果之间的关系：(a) 基于PLCM的密码系统的像素变化率结果，(b) 基于PLCM的密码系统的UACI结果，(c) 基于2DLASM的密码系统的像素变化率结果，(d) 基于2DLASM的密码系统的UACI结果。

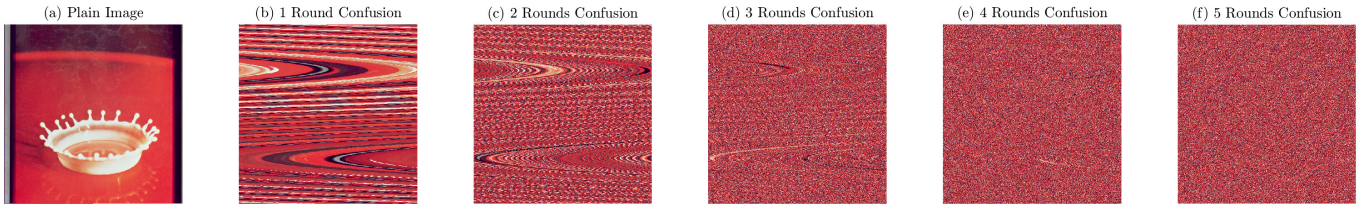

图10：混淆操作的结果。(a) 明文图像（水花），(b)-(f) 经过不同轮数混淆操作后的图像，。

6. 参数设置

根据加密过程，显然，辅助线程数n 以及混淆与扩散轮数r会显著影响已部署密码系统的加密速度和统计性能。在本文进行的所有实验中，笔记本电脑、个人电脑和工作站的n 分别设为 8、12 和 32，且r 设为 5。本节将分析采用这些设置的原因。

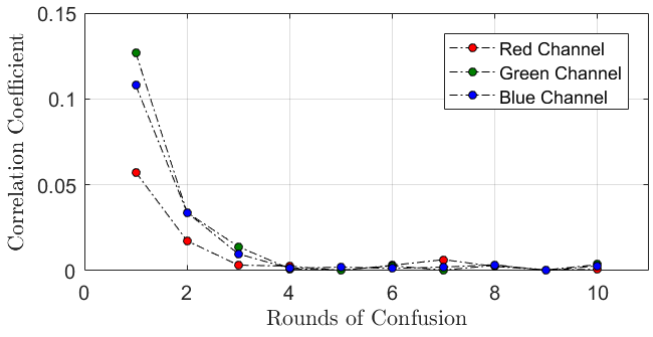

图11：明文图像与经过不同轮数混淆操作后的图像之间的相关系数。

我们使用不同数量的辅助线程，在三个硬件平台上对大小为 576×576 的明文视频进行加密。辅助线程数量与平均加密时间之间的关系如图 8所示。将 n 设为中央处理器支持的最大线程数并非必要，但可以确保所部署的密码系统达到最快的加密速度。

扩散与混淆操作旨在提高密码系统对明文的敏感性，并降低明文图像之间的相关性。因此，我们生成字节序列，对一幅明文图像执行 10 轮扩散操作，随机选择一个像素并更改其值，再使用相同的字节序列对修改后的图像执行 10 轮扩散操作，然后计算生成的密文图像之间的 NPRC 和 UACI。结果绘制于图 9中。类似地，我们对一幅明文图像执行 10 轮混淆操作。明文图像以及经过 1 至 5 轮混淆操作后的置乱图像如图 10所示。明文图像与经过不同轮混淆操作后的置乱图像之间的相关系数绘制于图 11中。当 r 等于 5 时，显然，像素变化率、UACI 和相关系数分别达到上下界。

7. 与以往工作的比较

如上所述，现有算法可分为全视频加密和选择性视频加密。显然，所提策略属于前一类。因此，本节将所采用的密码系统与近期发表的几篇全视频加密论文进行比较。结果如表 9所示。对于部分选定的研究，使用不同尺寸的视频来评估加密速度。表中列出了最大尺寸视频的加密速度。

随着信息科学和硬件的发展，用户对安全性提出了更高的要求。因此，几乎所有最新研究都采用混淆-扩散架构。然而，据我们所知，这些研究为了提高计算效率，仅包含一轮混淆和扩散操作，无法满足实时加密的要求，即视频帧的平均加密时间（毫秒）小于 1000 / 帧率。因此，与以往工作相比，所提策略具有以下优势：

i) 实现了实时视频加密，为相关应用提供了可行的解决方案，并为相关研究指明了新的方向。ii) 借助并行计算技术，证明了基于多轮混淆-扩散架构的实时视频加密是可行的，从而使实时视频加密的安全性达到图像加密的水平。iii) 适用于多种混淆方法、扩散方法和不同的混沌映射。

iv) 可通过软件和硬件（如 ARM、FPGA 等）轻松实现。

8. 结论

总之，本文提出了一种基于并行计算技术的实时混沌视频加密策略。该策略利用多个线程同时对相应子帧执行混淆和扩散操作，以提高计算效率。为评估该策略的性能，我们选取了两种混沌映射来构建密码系统。加密速度评估证明，该策略显著提升了加密速度，并能在不同硬件平台上实现实时视频加密。统计分析和安全性分析表明，所部署的密码系统具有出色的统计特性，并且能够抵御攻击、噪声和数据丢失。据我们所知，与以往的研究相比，所提策略实现了最快的加密速度，并首次基于多轮混淆-扩散架构实现实时视频加密。本文分别为相关应用和研究提供了一种可行的解决方案和一条新路径。

CRediT 作者贡献声明

DongJiang: 概念构思、方法论、撰写初稿、软件、监督。**ZhenYuan:** 调查、方法论、软件。**Wen-xinLi**：调查、软件。**Liang-liangLu:** 概念构思、方法论、撰写审阅与编辑、监督。

竞争利益声明

作者声明，他们不存在已知的竞争性经济利益或可能被认为会影响本文所报告工作的个人关系。

表9：与先前研究的速度比较

算法	r_c	r_d	CPU	内存	视频大小	CS	平均执行时间（毫秒）	RTE
参考文献 [17]	1	1	奔腾4@3.2GHz	4GB	240 × 360	RGB	1059（8维映射） 2314（12维映射） 1357（Ikeda延迟微分方程）	× × ×
参考文献 [18]	1	1	–	–	240 × 360	RGB	472	×
参考文献 [19]	1	1	英特尔酷睿i3-2130@3.4GHz	4GB	352 × 288	YUV	5960†	×
参考文献 [20]	1	1	英特尔酷睿 i5-2400@3.1GHz	4GB	–	RGB	408	×
参考文献 [21]	1	1	英特尔酷睿 i7-8750H@2.2GHz	16GB	352 × 288	RGB	260	×
参考文献 [22]	1	1	英特尔酷睿 i5-6200U@2.3GHz	8GB	360 × 240	RGB	5497	×
Ours	5	5	Intel Core i5-1135G7@2.4GHz	8GB	480 × 480	RGB	36.26 (PLCM) 35.49 (2DLASM)	✓ ✓
			Intel Core i7-8700@3.2GHz	32GB	576 × 576	RGB	39.92 (PLCM) 40.67 (2DLASM)	✓ ✓
			Intel Xeon Gold 6226R@2.9GHz	64GB	768 × 768	RGB	36.56 (PLCM) 36.23 (2DLASM)	✓ ✓

r_c：混淆轮数；r_d：扩散轮数；CS：色彩空间；AET：平均加密时间；RTE：实时加密（AET ≤ 1000 / 帧率）；–：未说明；†：参考文献 [19]，中未描述准确的平均加密时间，但参考文献 [21] 中给出了该时间。

数据可用性

数据可应要求提供。

致谢

本研究得到中国国家重点研发计划（2017YFA0303704）、国家自然科学基金（12274233, 62273001）、安徽省重大科研项目（KJ2021ZD0005）、安徽省编制科研计划项目（2022AH040020）、安徽省优秀科研创新团队项目（2022AH010005）以及安徽省高校协同创新项目（GXXT-2021-091）的资助。

参考文献

[1] Pareek NK、Patidar V、Sud KK。使用混沌逻辑斯蒂映射进行图像加密。Image Vision Comput 2006；24：926-934。[2] Fang P、Liu H、Wu C、Liu M。基于混沌系统的图像加密算法综述。Visual Comput 2022；2022：1-29。[3] Zhang Y、Wang B。基于干涉的光学加密。Opt Lett 2008；33：2443-2445。[4] Zhang Y、Wang X。基于混合线性–非线性耦合映像格子的对称图像加密算法。Inform Sciences 2014；273：329-351。[5] Tedmori S、AI-Najdawi N。基于Haar小波变换的图像密码算法。Inform Sci 2014；269：21-34。[6] Chai x、Gan Z、Chen Y、Zhang Y。基于压缩感知的视觉安全图像加密方案。Signal Process 2017；134：35-51。[7] Luo Y、Zhou R、Liu J、Cao Y、Ding X。一种基于分段线性混沌映射和超混沌映射的并行图像加密算法。Nonlinear Dynam 2018；93：1165-1181。[8] Zia U、McCartney M、Scotney B、Martinez J、AbuTair M、Memon J。关于在空间域、变换域和时空域中使用混沌映射的图像加密技术综述。Int J Inf Secur 2022；21：917-935。[9] Chen J、Zhu Z、Fu C、Yu H。一种采用新型基于像素置换的置乱方法的快速图像加密方案。Nonlinear Dynam 2014；77：1191-1207。[10] Wong K、Kwok B、Yuen C。一种用于基于混沌的图像加密的高效扩散方法。Chaos Soliton Fract 2009；41：2652-2663。[11] Matthews R。论“混沌”加密算法的推导。Cryptologia 1989；13：29-42。[12] Massoudi A、Lefebvre F、De Vleeschouwer C、Macq B、Quisquater JJ。图像和视频选择性加密综述：挑战与展望。Eurasip J Inf Secur 2008；2008：179290。[13] Liu F、Koenig H。视频加密算法综述。Comput Secur 2010；29：3-15。[14] Chiaraluce F、Ciccarelli L、Gambi E、Pierleoni P、Reginelli M。一种新型混沌视频加密算法。IEEE T Consum Electr 2002；48：838-844。[15] Lian S、Sun J、Wang J、Wang Z。一种混沌流密码及其在视频保护中的应用。Chaos Soliton Fract 2007；34：851-859。[16] Yang S、Sun S。一种基于DCT域混沌映射的视频加密方法。Prog Nat Sci 2008；18：1299-1304。[17] Valli D、Ganesan K。使用混沌映射和Ikeda时滞系统进行混沌视频加密。Eur Phys J Plus 2017；132：1-18。[18] Kumar RR、Ganeshkumar D、Suresh A、Manigandan K。一种基于一维混沌映射的单轮视频加密方案。载于：2019年第5届国际先进计算与通信系统会议；2019：439-444。[19] Li x、Yu H、Zhang H、Jin X、Sun H、Liu J。基于超混沌系统的视频加密。Multimed Tools Appl 2020；79：23995-24011。[20] Yasser I、Mohamed A、Samra AS、Khalifa F。一种用于安全多媒体通信的基于混沌的加密/解密框架。Entropy 2020；22：1253。[21] Hosny KM、Zaki MA、Lashin NA、Hamza HM。使用块置乱和多密钥生成的快速彩色视频加密。Visual Comput 2022；2022：1-32。[22] Dua M、Makhija D、Manasa PYL、Mishra P。一种基于三维混沌映射-余弦变换的视频加密与解密方法。Open Comput Sci 2022；12：37-56。[23] Hamidouche W、Farajallah M、Sidaty N、Assad S、Deforges O。基于可扩展HEVC扩展中混沌系统的实时选择性视频加密。Signal Process: Image Commun 2017；58：73-86。[24] Chen H、Tanougast C、liu Z、Blondel W、Hao B。基于改进Chirikov映射和旋转变换的光学高光谱图像加密。Opt Laser Eng 2018；107：62-70。[25] Fu C、Chen J、Zou H、Meng W、Zhan Y、Yu Y。一种采用改进扩散策略的基于混沌的数字图像加密方案。Opt Express 2012；20：2363-2378。[26] Zhong J、Jiang D、Huang Q、Cao Y。一种自更新数字混沌流密码。Int J Mod Phys 2018；29：1850074。[27] Hua Z、Zhou Y。使用二维Logistic调整正弦映射进行图像加密。Inform Sci 2016；339：237-253。[28] Li c、Lin D、Feng B、Lv J、Hao F。基于信息熵的混沌图像加密算法的密码分析。IEEE Access 2018；6：75834-75842。[29] Man Z、Li J、Di X、Sheng Y、Liu Z。一种基于神经网络和混沌的双图像加密算法。Chaos Soliton Fract 2021；152：111318。[30] Kwok HS、Tang WKS。一种基于有限精度表示混沌映射的快速图像加密系统。Chaos Soliton Fract 2007；32：1518-1529。[31] Zhang X、Zhao Z、Wang J。基于循环替换盒和密钥流缓冲区的混沌图像加密。Singal Process 2014；29：902-913。[32] Ye G、Wong KW。一种基于广义Arnold映射的高效混沌图像加密算法。Nonlinear Dynam 2012；69：2079-2087。[33] Rhouma R、Meherzi S、Belghith S。基于OCML的彩色图像加密。Chaos Soliton Fract 2009；40：309-318。[34] Belazi A、Khan M、El-Latif AAA、Belghith S。高效密码系统方法：S盒与基于置换替代的加密。Nonlinear Dynam 2018；87：337-361。[35] Wu Y、Zhou Y、Saveriades G、Agaian S、Noonan JP、Natarajan P。用于图像随机性的局部香农熵度量及统计检验。Inform Sci 2013；222：323-342。[36] Artiles JAP、Chaves DPB、Pimentel C。使用分组密码和混沌序列进行图像加密。Signal Process 2019；79：24-31。[37] Xian Y、Wang X。分形排序矩阵及其在混沌图像加密中的应用。Infrom Sci 2021；547：1154-1169。[38] Rukhin A、等人。面向密码学应用的随机数和伪随机数生成器统计测试套件。NIST特别出版物800-22，修订版1.a，2010。[39] Wang X、Teng L、Qin X。一种新型基于混沌的彩色图像加密算法。Signal Process 2012；92：1101-1108。[40] Alvarez G、Li S。基于混沌的密码系统的一些基本密码学要求。Int J Bifurcat Chaos 2006；16：2129-2151。[41] Khan M、Massod F。一种基于多个离散动力学映射的新型混沌图像加密技术。Multimed Tools Appl 2019；78：26203-26222。[42] Luo Y、Yu J、Lai W、Liu L。一种基于改进Baker映射和Logistic映射的新型混沌图像加密算法。Multimed Tools Appl 2019；78：22023-22043。[43] Mansouri A、Wang X。一种新型一维正弦幂混沌映射及其在新型图像加密方案中的应用。Inform Sci 2020；520：46-62。[44] Talhaoui MZ、Wang X。一种新型分数阶一维混沌映射及其在高速图像加密中的应用。Infrom Sci 2021；550：13-26。[45] Chen H、Bai E、Jiang X、Wu Y。一种基于改进的六维超混沌系统的快速图像加密算法。IEEE Access 2022；10：116031-116044。[46] Azzeh J、Zahran B、Alqadi Z。椒盐噪声：影响与去除。JOIV: Int J Inform Visual 2018；2：252-256。[47] Gan Z、Chai X、Han D、Chen Y。一种基于三维位平面置换的混沌图像加密算法。Neural Comput Appl 2019；31：7111-7130。