DOI:10.19651/j.cnki.emt.2417614

基于国密 SM2 的 H.264 视频加密方案设计*

郝 洋[1] 周 骅[1] 王代强[1,2]
(1.贵州大学大数据与信息工程学院 贵阳 550025;2.贵州民族大学物理与机电工程学院 贵阳 550025)

摘 要:本文针对传统视频加密系统针对大数据量加密无法兼顾实时性与安全性的问题,设计并实现了基于 SM2 算法视频加密方案。该方案基于国密 SM2 算法以及基于"xoshiro256ss"伪随机数生成器生成的轻量级流密码,使用混合加密以及伪"一次一密"保障数据的安全性。该方案通过 RTSP 协议实现设备与流媒体服务器的数据交互,能够通过 QT 对视频数据进行解密播放,为视频监控加密无法同时兼顾安全性与实时性的问题提供了一个有效解决途径。经过测试,该系统对 I 帧加密时间为 3 ms 左右。

关键词:SM2;流密码;选择性加密;混合加密;伪"一次一密"
中图分类号:TN37 **文献标识码**:A **国家标准学科分类代码**:510.40

Design of H.264 video encryption scheme based on national security SM2 algorithm

Hao Yang[1] Zhou Hua[1] Wang Daiqiang[1,2]
(1. School of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China;
2. School of Physics and Mechatronic Engineering, Guizhou Minzu University, Guiyang 550025, China)

Abstract: This article addresses the issue of traditional video encryption systems being unable to balance real-time and security in encrypting large amounts of data, and designs and implements a video encryption scheme based on the SM2 algorithm. This scheme is based on the State Secrets SM2 algorithm and a lightweight stream cipher generated by the "xoshiro256ss" pseudo-random number generator. It uses hybrid encryption and "one-time encryption" to ensure data security. This solution achieves data exchange between devices and streaming media servers through the RTSP protocol, and can decrypt and play video data through QT, providing an effective solution to the problem of video surveillance encryption that cannot simultaneously balance security and real-time performance. After testing, the system has an encryption time of about 3 ms for I-frames.

Keywords: SM2; stream cipher; selective encryption; mixed encryption; pseudo "one time one password"

0 引 言

视频监控技术因其广泛的应用领域而备受瞩目,如公共安全、交通管理、智能家居等。然而随着高清视频与图像技术的普及,人们在使用这些技术时也面临着非法查看、盗取和恶意篡改等安全风险。因此,如何在保障视频数据安全的同时,确保其实时性和可用性,成为当前研究的重要课题。

随着技术的不断进步和需求的不断变化,现如今视频加密主要有完全加密以及部分加密两大类,区别在于所加密数据的完整与否。目前国外采用 AES 算法较多,而基于国密算法的方案较少,文献[1]中提出 SM2 算法对视频文件进行了加密,并取得了良好的加密效果,文献[2]提出了仅使用 SM2 算法来加密 H.264 格式的视频,文献[3]提出了 SM2 以及 SM4 对本地视频进行加密,具有较好的安全性,但是这三种方案在加密时间上仍存在进一步优化的空间,当数据量过大时,加密时间也会相应增大。文献[4]提出 SM4 算法和 ARM_NENO 技术进行加速计算,虽然提高了加密速度,但其安全性仍然受限于对称加密算法的特性,未对密钥进行保护。文献[5]提出了一种基于国密标准的部分加密方法,对 H.264 中关键语义元素 IPM、MDV 和 RC 进行选择性加密,但对每一帧都提取相应的元素,需要花费额外时间。文献[6]提出了一种基于 SM4 的选择性视频加密算法,但未对密钥进行安全管理。文献[7-8]提出的

收稿日期:2024-12-12
*基金项目:国家自然科学基金(62272123)项目资助

混沌加密算法也常用于加密视频帧，但是此算法另外需要分离视频帧以及合成视频，因此该方案在数据量过大时，加密时间也会相应增加。

基于以上分析，本文旨在设计一个基于国密算法的实时视频加密方案，以满足视频监控的实时性和安全性需求。该方案将充分利用高性能硬件的算力，基于真随机数种子设计轻量级流密码、混合加密、选择性加密以及伪“一次一密”加密方案，以实现高安全性和实时性的平衡。具体而言，该方案基于国密 SM2 算法这类国内安全性极高的加密方法，构建软硬协同加密的平台，通过解决视频数据加密中的实时性和安全性问题，为视频监控技术的广泛应用提供有力保障。

1　基础知识

1.1　国密 SM2 算法

SM2 算法是我国国家密码管理局发布的一种椭圆曲线公钥密码算法[9]，主要包括数字签名、密钥交换以及数据加解密 3 个功能。

用户可以生成一对公钥和私钥，私钥 d 则是通过选择一个在椭圆曲线定义域内的随机数[10]，范围在 $[1,n-1]$ 之间，n 为椭圆曲线基点 G 的阶数，其长度为 32 字节。公钥 P 是通过椭圆曲线上的基点 G 与私钥进行点乘运算 $P=d\times G$ 得到[11]，其长度为 64 字节。

当使用 SM2 加密数据时生成一个随机数 k，并计算椭圆曲线点 $C_1=k\times G$，之后计算共享点 $(x_1,y_1)=k\times P$，并且使用密钥派生函数来生成密钥 t，$t=KDF(x_1\parallel y_1,len)$，$len$ 是待加密数据的长度；之后将 t 与明文 A 进行异或计算后得到密文数据 C_2，并计算哈希值 $C_3=Hash(x_1\parallel A\parallel y_1)$，最后得到密文 $C=C_1\parallel C_2\parallel C_3$。解密过程是加密过程的逆运算，使用私钥 d 计算共享点 $(x_1,y_1)=d\times C_1$，并通过 KDF 生成密钥 t；之后对 C_2 进行异或计算得到明文数据 A；最后对 C_3 进行哈希验证，验证数据的完整性。

SM2 算法的安全性基于椭圆曲线离散对数问题[12]，具有抵抗各种攻击和攻击模型的能力，具有极高的安全性。

1.2　H.264 视频帧格式

H.264 视频编码标准广泛应用于视频压缩，其基本单元是“宏块”，每个宏块由多个像素块组成。在编码过程中，视频帧由多个宏块组成[13]，如图 1 所示，这些视频帧主要有 3 种类型。

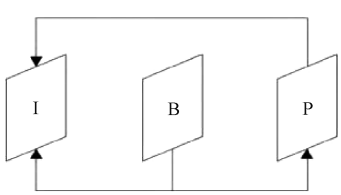

图 1　视频帧结构

Fig. 1　Video frame structure

I 帧（关键帧）：完全自包含的帧，包含完整的图像数据，不依赖于其他帧，通常用于场景的开始或快速切换时。

P 帧：通过预测前一帧的内容并编码差异来减少数据量，仅包含与之前的 I 帧或 P 帧的差异。

B 帧：通过参考前后两帧进行预测，从而进一步减少编码数据量。B 帧通常提供比 I 帧和 P 帧更高的压缩比，但其编码和解码的复杂度也相对较高。

H.264 标准通过多种技术提高视频流的压缩效率，包括帧内预测、帧间预测、变换和量化、去块滤波器以及熵编码。这些技术使 H.264 能够实现高效的视频压缩，同时保持良好的视频质量。

2　数据加密方案设计

2.1　伪随机数生成器

xoshiro256ss 伪随机数生成器（pseudo-random number generator，PRNG）由 David Blackman 和 Sebastiano Vigna 设计[14]，如算法 1 所示。该生成器具有非常高的随机性质量，并且具有非常好的速度性能，生成随机数的时间复杂度低于纳秒级别。

算法 1
输入：真随机数种子
输出：序列密码
a)　局部变量 S ← 状态变量 state
b)　计算 res=rol64(S)
c)　计算 t=S[1]左移 17 位
d)　更新状态变量 S
e)　S[2]=S[2] xor S[0]
f)　S[3]=S[3] xor S[1]
g)　S[1]=S[1] xor S[2]
h)　S[0]=S[0] xor S[3]
i)　S[2]=S[2] xor t
j)　S[3]=rol64(S[3],45)
k)　输出 res

PRNG 实现了一个状态更新机制，每次调用都会修改状态变量 s 的值。尽管其具有很高的速度和质量，但它并不是一个真正的真随机数生成器，因为它的输出是可以预测的。因此在本方案中，使用一个外部的真随机数源来初始化 PRNG 的状态，以获得更好的随机性。

2.2　流密码设计

本方案流密码种子使用 Linux 系统中的特殊文件/dev/random，它是加密安全随机数生成器（cryptographically secure pseudo-random number generator，CSPRNG）其中的一种，它生成的随机数质量非常高，被认为是足够安全的。它使用的是 Linux 内核中的随机数生成器，如图 2 所示，它结合了熵池（收集系统和用户活动产生的随机性）和

伪随机算法来生成随机数[15]。尽管它不是基于物理过程的“真随机数生成器”,但它生成的随机数序列具有很高的随机性和不可预测性。

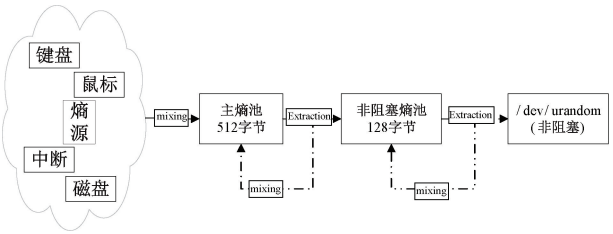

图 2　LRNA 结构

Fig. 2　LRNA structure

本文使用视频帧中提取的 256 位随机数与真随机数进行计算,得到伪随机数种子。因为连续两帧的视频数据通常不会完全相同,即使是在静态场景下,由于压缩算法和编码器的工作原理,视频帧也会略有变化。这样的设计首先确保了每次使用的密钥都是随机生成的,其次为了避免随机数种子生成过程中出现规律性数据,起到种子初始化的作用,确保序列密码生成器能够正常运行,具体流程如下:

Step1)得到编码后的数据 *src* 时,需要对数据进行判断,获取到关键帧中的 0X67 字节所在的偏移量 α。

Step2)根据该偏移量 *src*[α] 获取 256 位的视频编码数据 *src*[$\alpha+32$]。

Step3)将得到的数据 *src*[$\alpha+32$] 与真随机数进行异或计算,得到 256 位的伪随机数种子。

Step4)将该伪随机数种子输入到函数 xoshiro256ss()。

Step5)计算一个临时值 *result*,将 64 位 *s*[1] * 5 之后循环左移 7 位,之后再乘 9。

Step6)将 256 位数据分入 4 个元素数组:*s*[0],*s*[1],*s*[2],*s*[3]。

Step7)计算一个新的状态值 t,$t = s[1] << 17$。

Step8)通过 *s*[2] 和 *s*[0]、*s*[3] 和 *s*[1]、*s*[1] 和 *s*[2]、*s*[0] 和 *s*[3] 进行异或计算,从而更新 *s*[2] 和 *s*[3]。

Step9)将 64 位 *s*[3] 循环左移 45 位以更新 *s*[3]。

Step10)将 step5)中的 *result* 作为 64 位的伪随机数进行输出,之后的步骤用于状态更新,以确保下一次调用时生成不同的随机数。

之所以选用 *src*[α] − *src*[$\alpha+32$],是因为在实际测试中,不同型号的摄像头由于编码格式有着细微的不同,因此从相应关键帧读取到 0X67 字节所在的偏移量也不相同,所以偏移量呈现一个动态值,字节 0X67 是关键帧的标志位。

如果直接使用真随机数作为流密码,那么每次加密都需要一个新的、足够长的随机数序列。对于实时视频流来说,频繁地更新和分发大量的真随机数据加密会增加计算资源的负担,尤其是在计算资源有限的应用环境中。而基于种子生成流密码的方法更加高效,可以快速生成所需长度的流密码,因此使用产生的真随机数作为流密码基础密钥。如图 3 所示,通过真随机数发生器生成的随机数作为伪随机数发生器的种子传入伪随机数发生器,产生后续的序列密码,提升密码的安全性与高效性。

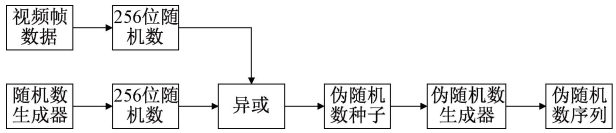

图 3　流密码设计流程

Fig. 3　Flow chart of stream cipher design

2.3　混合加密算法设计

如图 4 所示,使用 SM2 对 256 位的流密码进行加密计算,由 2.2 节可以生成的流密码加密视频编码数据,最后得到真随机数密文以及视频加密后的数据,即可进行数据传输。并且采用异或的计算方式降低计算量,尽管异或运算简单,但是序列密码的抗攻击能力很强。

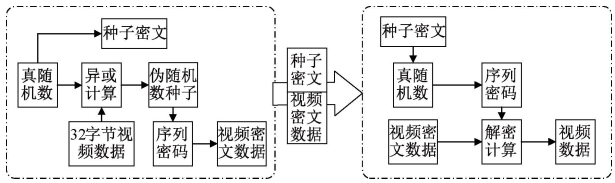

图 4　混合加解密流程

Fig. 4　Mixed encryption and decryption flow chart

加密流程具体实现如算法 2 所示,将这两部分密文数据进行合并,将序列密码的基础密钥密文和关键帧的加密数据进行组合,将数据一起通过网络信道传输到客户端,确保密钥的安全性和有效的密钥管理分发。

算法 2:

输入:data,data_len

输出:data_out

a)　if I=1 then

b)　　申请 Buffer

c)　　读取 SM2_pubkey

d)　　aa ← 真随机数种子

e)　　计算流密码 S ←xoshiro256ss(aa^data (α,$\alpha+32$))

f)　　计算 SM2_data ← SM2_encrypt(aa)

g)　　for i from to data_len do

h)　　　计算 data_out ← S^data

i)　　End

j)　Else

k)　　计算流密码 S* ← data(32)

l)　　for i from to data_len do

m)　　　计算 data_out ← S*^data (α,$\alpha+32$)

n)　　End

o)　输出 data_out、SM2_data

之后解密端进行数据拆解后得到序列密码的密文和经过加密的视频编码数据，将序列密码的密文使用对应的 SM2 私钥进行数据解密，得到所需要使用的流密码基础密钥，接着使用该基础密钥输入到伪随机数生成器得到流密码，流密码与视频数据进行计算得到原始明文数据。

3　视频加密方案设计

3.1　加密内容

如图 5 所示，本方案对 H.264 视频编码后的内容进行加密，而非对图像本身进行加密处理。

图 5　加密内容流程

Fig. 5　Encryption content flow chart

首先未加密的 H.264 视频流包含了一系列网络抽象层(network abstraction layer，NAL)单元作为 H.264 流的构建块[16]，这些单元包含了视频帧的编码信息。其次未加密的 H.264 视频流可以直接被 H.264 解码器解码为原始图像帧，加密后的 H.264 视频流必须先经过解密才能被解码器正确处理。如果没有正确的密钥，解码器将无法正确解析和解码视频流，导致原有的像素格式遭到破坏，无法生成有意义的图像。

3.2　选择性加密方案

为进一步提高加密系统的实时性，通过仅对部分关键帧进行混合加密。如图 6 所示，在接收到视频数据后，仅对 I 帧的数据进行混合加密，本文对非关键帧也进行了相应处理，其余帧使用对应视频帧提取的 256 位数据用作流密码进行异或计算进行加密。采用这样的设计可以在不损失安全性的情况下进行计算资源的合理分配，降低计算量，从而完成对所有帧数据的处理。因为即使攻击者能够访问非加密的中间帧，也无法还原整个序列，因为关键帧的信息丢失会导致序列的完整性受到影响。

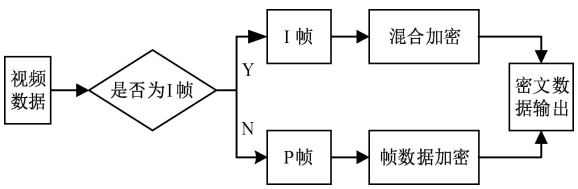

图 6　选择性加密流程

Fig. 6　Selective encryption flow chart

3.3　伪“一次一密”功能实现

本次加密方案中结合了伪“一次一密”的密钥更新机制，以确保数据安全性。具体而言，在加密过程中，采取了非关键帧之间种子基础值保持不变，而序列密码则随之变化的策略。

关键帧时种子更新：当关键帧发生变化时，种子 S 也随之更新。假设新的种子为 S_i，那么新关键帧的序列密码将基于 S_i 生成，设 f 是伪随机函数，I 为对应帧的 32 字节数据，那么第 i 帧的序列密码 K_i 可以表示为：

$$K_i = f(S_i, I) \tag{1}$$

由于 i 的变化，K_i 也是不同的。这样可以确保每个关键帧都有一个对应的新种子，从而生成独特的流密码。

非关键帧之间的序列密码变化：即种子基础值保持不变，通过使用相应帧的 32 字节数据用作序列密码 k，随着非关键帧的读取序列密码更新为 $k \to k'$，因此每次生成的序列密码都是不同的。

假设明文 A 的熵为 $H(A)$，密钥 K 的熵为 $H(K)$，由于 K 为伪随机序列，所以 $H(K) \approx n$(n 位的密钥)，根据信息论中的原理，当 A 和 K 相互独立时，密文 B 的熵可以表示为：

$$H(B) = H(A \oplus K) \tag{2}$$

对于两个独立且均匀分布的随机变量 A 和 K，异或操作 $A \oplus K$ 也是均匀分布的随机变量，因此密文 B 的熵等于密钥 K 的熵：

$$H(B) = H(K) = n \tag{3}$$

也就是说，密文 B 也是均匀分布的随机变量，没有任何可预测的信息。即使给定密文 B，明文 A 的条件熵 $H(A \mid B)$ 仍然等于明文的原始熵 $H(A)$，这意味着即使攻击者拥有密文 B，也无法从中获取关于明文 A 的任何额外信息。

通过种子更新和序列密码变化机制，确保每次使用的流密码是不可预测且唯一的，从而实现了“一次一密”的效果。这种设计不仅确保了每个关键帧都能生成独特的加密序列流密码，也在短周期内提高了真随机数的使用率。通过频繁地更新流密码，从而提升整体安全性。

4　视频加密系统测试

本文基于国密算法，构建起一个软硬件协同加密机制，国密算法安全且自主可控，系统方案设计如图 7 所示。

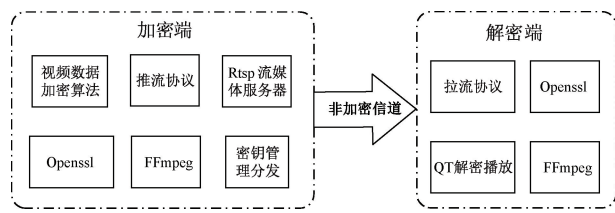

图 7　系统环境图

Fig. 7　System environment diagram

4.1　系统测试环境

基于上述方案设计，结合 RTSP 推拉流协议以及 QT 在嵌入式开发板上搭建起服务端和客户端的测试环境，系统测试如图 8 所示。

如表 1 所示，加解密端所使用的设备及软件库的名称、具体型号和版本号如下：

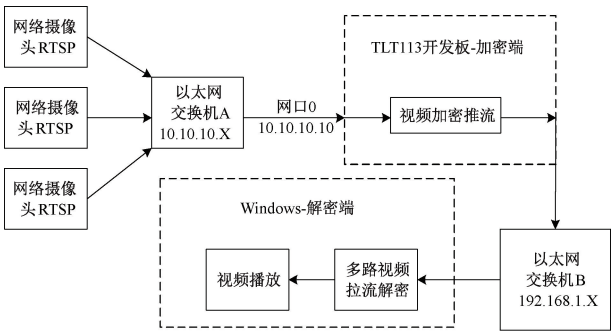

图 8　系统测试图

Fig. 8　System test diagram

表 1　具体开发环境表

Table 1　Specific development environment table

名称	型号
加密端 CPU	双核 ARM Cortex-A7
解密端 CPU	Intel(R) Core(TM) i5-3470
流媒体库	FFMPEG 4.2.3
密码库	OPENSSL 1.1.1k
服务器	RTSP 流媒体服务器

4.2　视频加密效果分析

1)主观效果分析

使用海康威视的网络摄像头采集 H.264 视频流进行测试。如图 9 所示,左图为原始视频画面,中间图为非正常解密播放画面,右图为正常解密画面。

图 9　加密效果对比

Fig. 9　Comparison of encryption effects

2)客观效果分析

结构相似度(structural similarity index measure,SSIM)是衡量两幅图像相似度的指标,能反映人眼对图像质量的感知。*SSIM* 的范围在−1～1,|*SSIM*|越接近 1,图像之间的相似度越高,当接近于或低于 0.1 时,图像可获得的有效信息就越少。其计算公式如下:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \tag{4}$$

其中,μ 表示图像的均值,σ 表示图像的方差,σ_{xy} 表示图像的协方差,C_1、C_2 是用于维持稳定的常数。

峰值信噪比(peak signal-to-noise ratio,PSNR)是通过计算两幅图像的均方误差并将其转换为信噪比,主要关注像素值的差异,对结构失真不敏感。其值越高,表示图像质量越好,反之则越低。若 *PSNR* 值小于 15 dB,则表明图像经过良好的加密处理。*PSNR* 的计算公式如下:

$$PSNR=10\lg\left(\frac{MAX_i^2}{MSE}\right) \tag{5}$$

其中,MAX_i 表示图像的最大像素值,*MSE* 表示原始图像与处理后图像之间的均方误差。

在运行加密系统时将 100 帧明文和密文保存本地,计算图像的 *SSIM*、*PSNR* 平均值,实验结果如表 2 所示。

表 2　SSIM 与 PSNR 数据表

Table 2　SSIM and PSNR data table

视频图像	数量	*SSIM*	*PSNR*/dB
I 帧	100	0.005 1	5.801 3
文献[3]		0.007 4	7.683 2
文献[7]		0.008 9	7.310 0

由表 2 可以看出,密文视频图像的结构相似度接小于 0.1,而峰值信噪比的数值均低于 15 dB。这表明密文视频图像与明文视频图像在结构上几乎没有相似性,显示出显著的失真程度。与文献[3]与文献[7]相比,本方案在明文图像于密文图像之间的数学关系上占优,加密效果较好。

4.3　安全性分析

1)直方图分析

在对经过加密的视频序列中,选择关键帧并对其进行灰度直方图分析。实验表明,经过编码后加密,原图的像素格式编码数据会被加密。如图 10 所示,如果在解密过程中出现异常,那么将无法恢复到原始的像素格式,解密失败后的图像原像素格式受到不可逆的破坏,这意味着即使有人获得了加密数据,也无法轻易地恢复出原始图像。实验结果显示,密文图像的像素分布与原文有较大差异,起到一个良好的像素置乱效果。

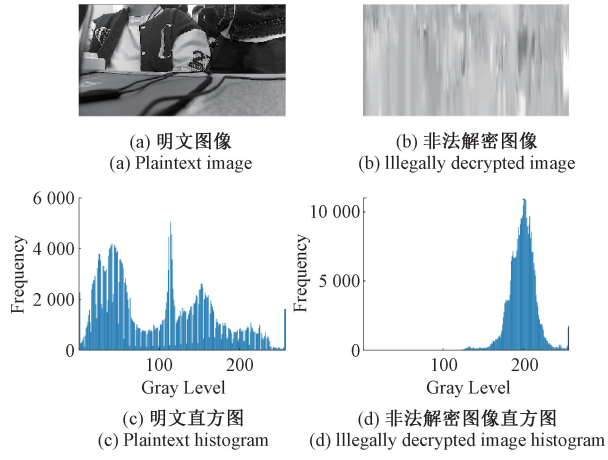

图 10　灰度直方图分析

Fig. 10　Analysis of grayscale histogram

2)相邻像素相关性

理想情况下密文图像具有很低的相邻像素相关性,说明其像素间的强相关性在加密后被打破,计算公式如下:

$$r_{xy}=\frac{cov(x,y)}{\sqrt{D(x)}\times\sqrt{D(y)}} \tag{6}$$

$$cov(x,y)=\frac{1}{N}\sum_{i=0}^{N}(x_i-E(x))(y_i-E(y)) \tag{7}$$

$$E(x)=\frac{1}{N}\sum_{i=0}^{N}(x_i),D(x)=\frac{1}{N}\sum_{i=0}^{N}(x_i-E(x))^2 \tag{8}$$

其中，r_{xy} 为相关系数，$cov(x,y)$ 为 x 与 y 的协方差，$\sqrt{D(x)}$ 与 $\sqrt{D(y)}$ 表示 x 与 y 的标准差。$E(x)$ 和 $D(x)$ 为 x 均值与方差，将 y 代入式(8)可得 $E(y)$ 和 $D(y)$，相邻像素相关性的值计算结果如表 3 所示。

表 3　图像相邻像素相关性数据表

Table 3　Image adjacent pixel correlation data table

图像	水平方向	垂直方向	对角方向
I 帧明文	0.962 4	0.948 5	0.927 5
I 帧密文	0.057 2	−0.014 8	−0.058 9

从表可以看出，明文图像在各个方向上的相邻像素具有很强的相关性，反观经过本系统处理过的密文图像的相邻像素相关性则相对较弱，相关系数绝对值均接近或低于 0.05，表明加密效果较好，接近随机分布。与文献[8]提出的基于混沌图像加密算法相比，虽然本方案的相邻像素相关性数据不够出色，但本方案更加适合用于需要实时处理的场景，基于混沌加密这类图像加密算法更适合用于本地重要视频文件的加密存储。

3)密钥空间分析

本方案设计的流密码是基于 256 位真随机数播种生成，其密钥空间大小为 2^{255}，是一个非常大的数值，使得暴力破解这样的方法几乎不可能有效，因此可以有效防止穷举攻击[17]。并且加密方案中还使用了混合加密算法以及密钥更新机制，对于每一帧可能的明文 data，都有 2^{255} 种可能的密文 DATA 与之对应。

此外，流密码的管理以及分发使用了国密 SM2 算法，该椭圆曲线公钥算法的私钥空间也为 2^{255}。并且国密 SM2 具有很高的理论安全性，能够很好的对流密码进行保护。

4)流密码安全性分析

(1)流密码 NIST 分析

将生成的流密码进行收集，通过 NIST 随机数评价软件对采集到的 230 万位流密码进行分析，如图 11 所示，主要的 8 个指标均大于 0.01，因此本文设计的流密码具有良好的随机性。

(2)流密码线性复杂度分析

BM 算法(Berlekamp-Massey)常用于计算线性反馈移位寄存器的最小长度和反馈多项式，本文将其用于计算二进制序列(流密码)的的线性复杂度，具体步骤如下：

①输入流密码序列：$S=s_0,s_1,s_2,\cdots,s_{n-1}$。

②初始化：设置初始的长度为 $L=0$，反馈多项式 $C(x)=1$，辅助多项式 $B(x)=1$，初始的差异 $d=1$。

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
   generator is </home/haoyang/linux/nist/sts-2.1.2/sts-2.1.2/random.txt>
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  1   2   0   3   2   5   0   3   2   2 0.350485     20/20      Frequency
  2   1   2   0   3   1   3   2   2   4 0.739918     20/20      BlockFrequency
  0   1   3   1   0   3   1   5   0   6 0.012650     20/20      CumulativeSums
  0   1   1   4   2   1   1   3   3   4 0.437274     20/20      CumulativeSums
  5   0   3   4   1   1   1   1   1   3 0.213309     19/20      Runs
  1   2   0   3   2   2   1   4   3   2 0.739918     20/20      LongestRun
  4   2   3   3   1   2   2   0   2   1 0.739918     20/20      Rank
  5   2   1   1   0   2   2   2   2   3 0.534146     19/20      FFT
```

图 11　NIST 分析数据图

Fig. 11　NIST analysis data graph

③迭代计算：对每个 i 从 0 到 $n-1$ 计算差异，其公式如下：

$$d_i=s_i+\sum_{j=1}^{L}c_j s_{i-j} \tag{9}$$

其中，c_j 是当前反馈多项式 $C(x)$ 的系数。如果 d_i 等于 0，那么就继续计算下一个数据，如果不为 0，则计算新的反馈多项式，计算公式如下：

$$C^*(x)=C(x)+d_iB(x)x^{i-m} \tag{10}$$

其中，m 为上一次更新的索引。如果 $2L\leqslant i$，则更新 $L=i+1-L$ 并且设置 $B(x)=C(x)$ 以及 $m=i$，之后更新 $C(x)=C^*(x)$。

④输出：最终 L 是流密码序列的线性复杂度，$C(x)$ 是最小的反馈多项式。

通过以上步骤，最终计算得到 256 位流密码的线性复杂度为 128，对于一个长度为 n 的序列，如果其线性复杂度 L 接近 $\frac{n}{2}$ 或更高，则说明该序列的线性复杂度较好，如果远小于 $\frac{n}{2}$，则不适合用于高安全性的应用，因此本文设计的流密码具有良好的线性复杂度。

4.4　加密效率分析

加密效率是指加密算法在单位时间处理数据的能力，在相同数据量下，高效的加密算法完成加解密的时间会更短。选取 I 帧以及 P 帧的测试结果如表 4 所示，密文的加解密效率良好，在监控系统中只引入了 5 ms 左右的延迟，满足安全性与实时性的要求，关键帧加密时间主要用于 SM2 加密流密码种子。加密 P 帧时间中二进制数据的异或计算带来的加密延迟以微秒为单位，即使数据量大量增加，本方案的加密时间也仅以微秒为单位上升，因此将视频帧数据量进行换算比较，相比较于文献[2]、文献[3]以及文献[5]，本方案具有更好的实时加密速度，有更高的加密效率。

4.5　图像质量分析

由于编码后进行加密，在理想情况下，编码后的加密与解码前的解密为对称操作，因此不会影响图像的质量。本文对比了密文图像、解密后的图像与原始图像之间的 $PSNR$，实验结果如表 5 所示。

通过对随机选取的 5 帧图像进行 PSNR 对比分析，该加密方案并没有过度损害解码后图像的质量。同时，加密

表 4　加密时间表

Table 4　Encryption schedule table

图像	大小/KB	加密/ms	解密/ms
I 帧	127.197 1	3.671 4	1.273 7
P 帧	10.403 2	0.052 2	0.051 6
文献[2]	101 856	361 480	未给出
文献[3]	6 144	1 000	1 000
文献[5]	IPM、MVD、RC	32	未给出

表 5　图像 PSNR 数据表

Table 5　Image PSNR data table

序号	密文图像/dB	解密图像/dB	原始图像/dB
1	5.924 1	35.472 5	36.668 1
2	5.517 2	37.535 2	37.578 4
3	5.734 1	40.136 8	41.716 8
4	6.137 1	39.432 8	41.142 9
5	5.894 1	37.719 4	39.428 4

操作导致密文图像的 *PSNR* 值显著下降,这说明加密成功地掩盖了图像信息,增强了数据的安全性。

5　结　　论

本文针对传统视频加密无法很好兼具安全性与实时性的问题,设计了一种实时 H.264 视频加密方案。

本文基于“xoshiro256ss”伪随机数生成器实现轻量级流密码生成器设计,适合在计算资源有限的环境中运行;此外基于国密 SM2 算法对流密码种子进行加密,并使用流密码对视频编码数据进行加密,不仅保证了数据的安全,还提高了加密效率,这样的混合加密方案设计在保障数据安全性的同时,也满足了视频监控的实时性要求;并且通过对 H.264 视频流中的关键帧进行混合加密,对非关键帧使用基于视频帧提取的数据作为流密码进行加密,在降低计算成本的同时保证了加密的安全性和完整性;最后在加密过程中,设计了关键帧之间流密码种子更新,非关键帧之间流密码种子不变,序列密码随之变化的方案,实现了伪“一次一密”的机制,通过频繁更新流密码,降低了攻击者预测加密序列的可能性,从而提升了整体安全性。

测试结果表明,系统加密 I 帧用时 3 ms 左右,解密用时 1 ms 左右;系统加密以及解密 P 帧用时 0.05 ms 左右。相比于传统的视频加密系统,该方案不但可以充分保障加密密钥以及视频数据的安全,而且可以有效的缩短了数据加密所需要的时间,在视频实时加密播放的安全性与延迟之间达到了一个良好的平衡点。

参考文献

[1] 沈哲林,熊显名. 基于国密算法 SM2 的视频文件加解密系统[J]. 工业控制计算机,2024,37(4):80-81,84.
SHEN ZH L, XIONG X M. Video file encryption and decryption system based on domestic algorithm SM2[J]. Industrial Control Computer, 2024, 37(4): 80-81,84.

[2] 王溪波,戚成烨,贾正锋. 基于国密算法的视频媒体文件加密效率提升技术[J]. 计算机系统应用,2024,33(2):43-53.
WANG X B, QI CH Y, JIA ZH F. Encryption efficiency improvement technology of video media file based on national secret algorithm [J]. Computer Systems & Applications, 2024, 33(2): 43-53.

[3] 林浩,黄一平,梁梓辰. 加密视频播放系统设计与实现及其安全性分析[J]. 电子设计工程,2024,32(12):150-156,161.
LIN H, HUANG Y P, LIANG Z CH. Design and implementation of encrypted video playback system and itssecurity analysis [J]. Electronic Design Engineering, 2024, 32(12): 150-156, 161.

[4] 韩超,周骅,赵麒. ARM NEON 和国密 SM4 的 H.264 视频加密[J]. 单片机与嵌入式系统应用,2023,23(3):60-63.
HAN CH, ZHOU H, ZHAO Q. H.264 video encryption based on ARM NEON and national secret SM4[J]. Integrated Circuits and Embedded Systems, 2023, 23(3): 60-63.

[5] 许盛伟,邓烨,刘昌赫,等. 一种基于国密算法的音视频选择性加密方案[J]. 信息网络安全,2023,23(11):48-57.
XU SH W, DENG Y, LIU CH H, et al. A selective encryption scheme for audio and video based on the national cryptographic algorithm[J]. Netinfo Security, 2023, 23(11): 48-57.

[6] 袁志民,朱春磊,吕成都. 基于 SM4 的选择性视频加密算法[J]. 通信技术,2019,52(8):1962-1966.
YUAN ZH M, ZHU CH L, LYU CH D. Selective video encryption algorithm based on SM4 [J]. Communications Technology, 2019, 52(8): 1962-1966.

[7] 刘博. 基于整数混沌的视频加密算法研究[D]. 北京:北京化工大学, 2020.
LIU B. Research on video encryption algorithm based on integer chaos[D]. Beijing: Beijing University of Chemical Technology, 2020.

[8] 曾祥秋,叶瑞松. 基于改进 Logistic 映射的混沌图像加密算法[J]. 计算机工程,2021,47(11):158-165,174.
ZENG X Q, YE R S. Chaotic image encryption algorithm based on improved Logistic map [J]. Computer Engineering, 2021, 47(11): 158-165, 174.

[9] 刘泽超,梁涛,孙若尘,等. 基于国密算法的 MQTT

安全机制研究与实现[J]. 计算机科学,2024,51(2):333-342.
LIU Z CH, LIANG T, SUN R CH, et al. Research and implementation of MQTT security mechanism based on national secret algorithm [J]. Computer Science, 2024,51(2):333-342.

[10] 尤文珠，葛海波. 利用多基数系统的高效椭圆曲线多标量乘算法[J]. 计算机工程,2021,47(2):182-187.
YOU W ZH, GE H B. Efficient algorithm for multi-scalar multiplication of elliptic curves using multi-base number system [J]. Computer Engineering, 2021, 47(2):182-187.

[11] 张吉鹏，黄军浩，于璇，等. 面向移动设备的国密 SM2 高效实现研究[J]. 电子学报,2023,51(12):3437-3443.
ZHANG J P, HUANG J H, YU X, et al. Research on efficient implementation of SM2 for mobile devices[J]. Acta Electronica Sinica, 2023,51(12):3437-3443.

[12] 唐亦昕，张英男. 基于 SM2 和 DNA 的图像加密算法[J]. 陕西科技大学学报,2025,43(1):203-210.
TANG Y X, ZHANG Y N, The image encryption algorithm based on SM2 and DNA [J]. Journal of Shaanxi University of Science & Technology, 2025, 43(1):203-210.

[13] 戴聪. 基于国密算法和模糊提取的多因素身份认证方案[J]. 计算机应用,2021,41(S2):139-145.
DAI C. Multi-factor authentication scheme based on national secret algorithm and fuzzy extractor [J]. Journal of Computer Applications, 2021, 41 (S2): 139-145.

[14] BLACKMAN D, VIGNA S. Scrambled linear pseudorandom number generators[J]. ACM Transactions on Mathematical Software(TOMS), 2021, 47(4): 1-32.

[15] 刘攀. 软件随机数发生器设计与实现技术研究[D]. 武汉:中国科学院大学,2020.
LIU P. Software random number generator design and implementation technology research [D]. Wuhan: University of Chinese Academy of Sciences,2020.

[16] 田野，郭弘，李岩，等. 司法鉴定视域下嵌入式 DVR 系统的数据恢复可行性探究[J]. 中国司法鉴定,2024(1):74-81.
TIAN Y, GUO H, LI Y, et al. Exploring the feasibility of embedded DVR systems data recovery under the perspective of forensic science[J]. Chinese Journal of Forensic Sciences, 2024(1):74-81.

[17] 孙夏晨，明鹏，李文石. 基于比特全置乱的超混沌图像加密算法[J]. 电子测量技术,2021,44(12):128-132.
SUN X CH, MING P, LI W SH. Hyperchaotic image encryption algorithm based on bit full scrambling[J]. Electronic Measurement Technology, 2021, 44(12): 128-132.

作者简介

郝洋，硕士研究生，主要研究方向为嵌入式系统设计。
E-mail:2075923494@qq.com

周骅(通信作者)，副教授，博士，主要研究方向为嵌入式系统设计、物联网安全。
E-mail:Zhouhua97@gmail.com

王代强，教授，博士，主要研究方向为电磁场与微波技术。