计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(2):43−53 [doi: 10.15888/j.cnki.csa.009389]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于国密算法的视频媒体文件加密效率提升技术①

王溪波[1], 戚成烨[1], 贾正锋[2]

[1](沈阳工业大学 信息科学与工程学院, 沈阳 110870)
[2](沈阳风驰软件股份有限公司, 沈阳 110167)
通信作者: 戚成烨, E-mail: qichengye@smail.sut.edu.cn

摘　要: 随着计算机网络和无线通信等技术的发展, 有关视频媒体文件的版权保护和信息安全问题日益成为人们关注的焦点, 对视频媒体文件加密是一种有效保护信息安全的方式, 传统的视频文件加密方法需要对视频媒体文件中所有的视频帧数据进行加密, 文件加密的效率较低, 加密过程比较耗时. 本文针对 H.264/AVC 视频帧的结构特点, 提出了一种基于国产 SM2 算法的视频媒体文件加密效率提升的方法, 该方法在加密视频媒体文件的过程中只加密视频数据中关键帧的 NALU Header 信息, 在检测到 H.264 分片的情况下同时也需要对 non-IDR Header 信息进行加密. 实验结果表明该方法可以在有效加密视频媒体文件的同时减少了加密所需的时间, 明显提升了视频媒体文件的加密效率.

关键词: 视频媒体文件; 信息安全; H.264/AVC 视频帧; 国产 SM2 算法; 加密效率提升

引用格式: 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.计算机系统应用,2024,33(2):43–53. http://www.c-s-a.org.cn/1003-3254/9389.html

Encryption Efficiency Improvement Technology of Video Media File Based on National Secret Algorithm

WANG Xi-Bo[1], QI Cheng-Ye[1], JIA Zheng-Feng[2]

[1](School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China)
[2](Shenyang Fengchi Software Co. Ltd., Shenyang 110167, China)

Abstract: With the development of technologies such as computer networks and wireless communication, copyright protection and information security issues of video media documents have become increasingly the focus of people's attention, and video media file encryption is a way to effectively protect information security. Traditional video file encryption methods need to encrypt all video frame data in video media files. The efficiency of file encryption is relatively low, and the encryption process is time-consuming. Therefore, a method for improving the efficiency of video media file encryption based on the Chinese SM2 algorithm is proposed according to the structural characteristics of H.264/AVC video frames. This method only encrypts the NALU Header information of the key frame in the encrypted video media during video media file encryption. In the case of detecting H.264 shards, it is also necessary to encrypt the non-IDR Header information. The experimental results show that the method can effectively encrypt video media files while reducing the time required for encryption, thus significantly improving the encryption efficiency of video media files.

Key words: video media file; information security; H.264/AVC video frame; Chinese SM2 algorithm; encryption efficiency improvement

① 基金项目: 辽宁省自然科学基金面上项目 (2022-MS-438); 辽宁省第二批揭榜挂帅科技攻关专项 (2022JH1/10800085); 辽宁省教育厅基本科研项目服务地方项目 (LJKFZ20220184); 2022 年度沈阳市科学技术计划 (22-316-1-07)
收稿时间: 2023-07-26; 修改时间: 2023-08-24; 采用时间: 2023-09-26; csa 在线出版时间: 2023-12-25
CNKI 网络首发时间: 2023-12-27

随着移动互联网与信息压缩技术的飞速发展, 视频媒体文件的传输与通信变得十分便捷, 大量有关个人、企业和政府的视频媒体数据需要产生、加工并经过互联网来传输与访问. 由于网络开放性与共享性的特点, 人们可以随时随地多设备对视频资源进行存取, 但这一趋势却给人们带来对视频媒体数据隐私和安全方面的担忧, 例如视频数据在传输过程中可能存在着隐私泄露、视频窃取、视频内容恶意篡改等安全隐患, 如何保证视频数据在网络传输和存储过程中的安全成为了一个热门的话题. 使用视频加密算法对视频媒体数据进行加密是一种有效的数据保护手段, 它可以将明文数据通过密钥和加密算法变成混乱无规律的密文数据, 从而达到保护视频媒体文件的目的, 通常视频媒体文件较大, 一部时长 2 h 的视频, MPEG-1 编码的视频大约是 1 GB 大小, 并且需要以实时或高帧率进行传输和播放, 如果使用传统加密算法 (如 DES, AES 和 IDEA 等)[1]加密视频文件, 通常是将视频媒体文件数据全部进行加密[2], 这样加密确实可以带来更高的安全性, 但需要的计算量也会非常大, 导致加密效率过低, 加密所需的成本也往往超出了视频媒体文件本身的价值. 因此, 在加密过程中需要考虑效率和性能, 以确保对视频数据进行加密和解密的时间开销在可接受的范围内, 为解决上述问题, 本文主要针对视频媒体文件加密效率提升技术进行研究.

视频媒体文件加密效率提升技术在各种场景和应用领域都有重要作用, 例如, 在流媒体视频服务领域, 视频流媒体平台需要对视频内容进行加密, 以保护版权、防止盗播和非法下载, 提高视频加密效率可以有效减少视频的解密延迟和缓冲时间, 从而提高视频播放质量, 改善用户体验; 对于企业视频安全而言, 企业内部视频资料的安全性对于保护商业机密和隐私至关重要, 提高视频加密效率可以在保护视频文件的同时不影响员工或合作伙伴的正常使用和访问; 在远程监控和视频会议方面, 远程监控系统和视频会议系统需要对视频流进行加密, 以保护视频传输过程中的隐私和机密信息. 高效的加密技术可以降低延迟, 提供稳定和安全的通信环境, 总体来说, 通过提升视频媒体文件加密效率, 可以在多个应用领域中实现更好的数据保护和提升用户体验.

在加密算法方面我国的研究已取得显著成果, 2010 年国产商用密码算法经国家密码管理局发布, 成为我国以及 ISO/IEC 国际密码标准, 其中国产 SM2 算法已在全国范围内进行了推广和普及[3], 涵盖了安全传输、身份认证等众多领域, 为系统安全提供了安全技术保障. 本文针对 H.264/AVC 视频帧的结构特点, 以及对目前普遍应用于视频加密的 DES, AES 和 IDEA 对称加密算法进行分析可知, 非对称加密 SM2 算法在安全性方面上比对称加密算法提供更高级别的安全性, 并且使用 SM2 算法可以有效避免由于使用国际密码技术所带来的安全风险, 即从算法层面上推动自主知识产权, 确保加密过程的安全性, 然而目前还没有相关研究将 SM2 算法应用到视频加密的领域中, 因此基于以上原因本文开创性地提出一种基于国密 SM2 算法的视频媒体文件加密效率提升的选择性视频加密方法, 分析实验结果可知该方法大幅度地缩短了加密时间, 在保证加密效果的同时提升了加密效率, 最终达到高效加密视频媒体文件的目的.

1 视频媒体文件加密

在过去的几十年中研究者们提出了许多针对 H.264/AVC 视频的加密方法[4], 不同的视频加密方法是针对不同的加密需求或目的而进行设计的, 如提高数据的安全性、实时性、保证加密文件的比特率和加密文件的格式兼容性等. 尽管如今关于文本加密算法的研究已经非常成熟, 但是如果将传统的文本加密算法 (如 DES, AES 和 IDEA 等) 直接应用到视频加密中, 虽然能够获得非常高的安全性, 但也存在计算复杂度高、加密效率过低和格式不兼容等缺点. 视频选择性加密 (selective encryption, SE)[5]是一种通过只加密视频数据流中的部分重要元素来达到降低计算复杂度并保持足够安全性的一种加密方法, 因为 SE 的部分在原始视频数据中只占小部分, 这使得 SE 和解密的成本相对于朴素算法加密也有了显著的降低, 对视频媒体文件的加密效率得到明显的提升, 因此可满足视频数据的实时性要求.

在视频编码中, 帧内预测模式 (intra-prediction model, IPM) 对于宏块内系数的编码十分关键, 运动向量也是 H.264/AVC 解码器中的宏块间系数进行预测与重建的重要数据, 因此许多文献[6–8]针对这两者进行加密以达到扰乱效果. Ahn 等[9]根据 I 帧的重要性, 提出了针对分割尺寸为 4×4 大小的 I 宏块的 IPM 进行异或加密的方法. 在 H.264/AVC 标准中, I 帧作为图像组 (group of

picture, GOP) 的首帧而携带大量数据, 并成为后续几乎所有 P 帧和 B 帧的参考基础, 因此, I 帧的加密效果不但会在帧内扩散, 还会明显地影响同一 GOP 内其余帧的画面质量, 具有较好的加密效果, 但所使用的异或加密方法易被破解, 导致加密后的视频数据安全性较低. 文献[10]提出了一种基于 H.264/AVC 的 SE 方案, 对 IPM、残差数据、帧间预测模式和运动矢量差 (MVD) 进行了加密, 该方案有着较高的加密效率和较低的传输延迟, 但是该方案依赖于加密算法和秘钥管理的可靠性, 否则可能会导致信息泄露. 文献[11]提出使用时空混沌加密模型对 IPM、MVD 以及熵编码语法元素进行加密, 通过对少量数据进行加密来实现高安全性加密, 还可以通过调整参数、初始化条件等方式产生不同的混沌序列, 从而实现多样化的加密过程, 这样可以增加攻击者的破解成本, 并且适应不同需求场景下的加密要求, 不足之处是时空混沌加密模型涉及到混沌映射、非线性动力学等复杂数学计算, 使加密和解密的运算复杂度更高, 在解密过程中, 需要进行大量的数学计算和迭代操作来恢复原始数据, 这对于硬件设备或者资源受限的场景可能带来挑战.

DCT 系数对于视频编码同样十分重要, 因此选择对 H.264/AVC 视频比特流的 DCT 系数进行加密也可以得到很好的加密效果. 在文献[12,13]中, 一种基于 MPEG-2 编码标准的加密方案被提出, 其加密对象是帧内宏块中 DCT 系数, 该加密方案利用一个控制因子可以灵活调节视频内容的扰乱程度, 且加密的 DCT 系数是编码中非常重要的数据, 故该算法安全性较高, 但该加密算法影响了压缩率且也只有有限的视频扰乱效果. Shahid 等[14]提出了一种针对 H.264/AVC 视频的快速加密方案, SE 非零系数后缀的符号位, 该方案具有格式兼容性且高效, 但只加密残差数据, 而纹理信息和运动信息仍然暴露在外, 因此安全性可能有所不足. 针对移动端特点的视频加密, Chung 等[15]提出一个使用 AES 加密 I 宏块中 DCT 系数和 P 宏块中 MV 的 SE 方案, 该方案虽然获得了很好的加密效果, 但是密文视频文件比特率增长明显, 导致解码复杂度增加, 增加传输开销.

根据上述相关工作可知, 现有更多视频安全研究只选择视频编码数据中的关键部分进行加密, SE 方案往往具有较好的加密效果, 由于现有的视频加密方法在安全性、计算复杂度和格式兼容性等方面存在一些问题, 尤其是一些加密方法加密效率较低, 因此本文将采用 SE 方案对视频媒体文件加密效率提升技术进行研究, 来完善以上研究中对视频数据加密方法中的一些不足.

2 视频媒体文件格式分析

本文主要是基于国密算法对视频媒体文件加密效率提升技术的研究, 在日常生活中我们常见的视频媒体文件格式有 MKV、AVI、MOV 和 MP4 等, 其中 MP4 格式的视频文件的特点是体积小占用空间少, 下载和传输起来方便, 并且能够在大多数播放器上播放, 具有较高的清晰度, 视频格式还可以轻易被转换成其他格式, 以便在特定的播放器上播放, 由于 MP4 文件具有以上这些优势, 所以在互联网领域被广泛应用, 本文也以 MP4 文件作为视频媒体文件的一个示例进行分析, 使用国密算法来对 MP4 文件的加密效率提升技术进行研究.

2.1 H.264/AVC 视频帧结构

现在常见的视频编解码格式主要有 H.264/AVC 和 H.265/HEVC[16–19]这两种, H.265 相比于 H.264 具有更高的压缩率, 可以实现更高的视频分辨率和更高的帧率, 此外, H.265 还具有更好的运动估计和色彩转换性能, 可以提供更清晰、更流畅的视频. 但是在一些应用场景下, H.264 可能仍然具有一定的优势, 比如低码率、低延迟的视频传输等, 并且 H.264 已经在市场上使用了相当长的时间, 具有广泛的兼容性和可用性, 由于 H.265 和 H.264 都使用 NALU 作为视频数据的传输单元, 只是在 NALU 头部长度和 NALU 类型字段有些不同, 所以本文选择 H.264/AVC 作为研究对象, 使用的方法也同样适用于 H.265/HEVC 压缩格式.

在 H.264/AVC 的 3 种视频编码中, I 帧表示关键帧, 可以理解为这一帧画面的完整保留[20], 解码时只需要本帧数据就可以完成, 是靠尽可能去除图像空间冗余信息来压缩传输数据量的帧内编码图像. P 帧是前向预测帧, 表示的是这一帧跟之前的一个关键帧 (或 P 帧) 的差别, 解码时需要用之前缓存的画面叠加上本帧定义的差别, 生成最终画面. B 帧是双向差别帧, 也就是记录的是本帧与前后帧的差别, 要解码不仅要取得之前的缓存画面, 还要解码之后的画面, 通过前后画面与本帧数据的叠加取得最终的画面. I 帧、P 帧、B 帧的关系如图 1 所示, 可见在 3 种类型中最重要的是 I 帧和 P 帧, B 帧依赖于 I 帧或 P 帧.

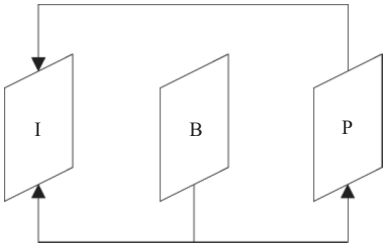


图 1 I、P、B 帧的关系图

2.2 MP4 加密可行性分析

MP4 文件本身是由一系列 Box 组成, 并且 Box 中也可以包含一系列子 Box, 文件中的媒体信息和媒体数据是分开存放的, 媒体数据保存在 Mdat Box 中, 由一系列音视频交织存放的块 (Chunk) 组成, 每个 Chunk 又由一系列类型相同且连续存放的样本 (Sample) 组成[21]. 在 MP4 载荷的 H.264 编码视频数据中, 一般情况下每一个 Sample 就是一个图像视频帧, 于是就会得出这样的结论, 如果对 H.264/AVC 视频帧进行加密, 那么实际上也会间接地对 MP4 文件进行加密, 因为 MP4 文件中的视频帧数据是来源于 H.264 编码中的视频帧, 这样 MP4 文件与 H.264/AVC 视频帧两者之间就建立了联系, 所以当视频帧经过加密后, 存储在 MP4 文件中的相应 Sample 和 Chunk 也会受到影响, 这将导致加密后的 MP4 文件无法正确解析和播放, 因为视频帧数据已被加密, 无法被正常解码和显示, 从而证明了 MP4 文件的加密可行性.

具体的加密原理如下, 在如图 2 所示的 H.264 码流分层结构中, Sample 由一个或多个 NALU 组成, 根据 H.264/AVC 语法, 一个 NALU 由一组对应于视频编码的 NALU 头部信息和一个原始直接序列负载 (RBSP) 组成, 根据 NALU 的头部信息可以判断这个 NALU 中承载的数据类型, 也就可以判断出该视频帧的类型, 视频序列中第 1 个视频帧是 I 帧, 而第 1 个前向预测帧 P 帧也参考的 I 帧, 双向差别帧 B 帧也依靠 I 帧, 所以本文提出使用国密算法来加密 I 帧的 NALU 头部信息的方法, 因为想要正确解析一帧数据, 必须先要正确解析构成它的 NALU 头部信息, 否则 I 帧无法被解析, 导致 P 帧和 B 帧也无法被解析, 所有视频帧的 Sample 出错, 由于在 MP4 文件中 Chunk 和 Sample 关系使得 Chunk 中包含的媒体数据无法被元数据正确描述, 并且只需加密 H.264/AVC 中少部分数据, 与传统加密算法相比, 计算量大幅度减少, 进而实现对视频媒体文件加密效率的提升, 符合视频加密轻量和高效型的研究方向.

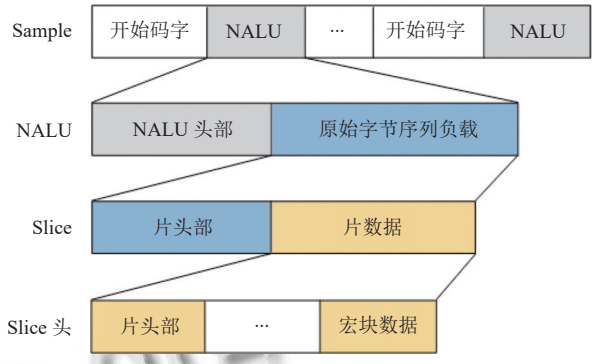


图 2 H.264 码流分层结构图

3 视频媒体文件加密效率提升技术研究

根据第 2 节对视频媒体文件格式的分析可知, 我们可以采用一种加密方式来对 H.264/AVC 视频帧中关键帧的 NALU Header 数据进行保护, 这种加密技术不仅能有效地加密视频媒体文件, 而且还能大范围地提高加密效率.

本节主要分为以下几个部分来介绍该加密方式, 首先, 在第 3.1 节中, 介绍了如何在 H.264/AVC 视频码流中找到关键帧的方法. 接下来, 在第 3.2 节讨论了在视频流获取和传输过程中可能出现的一些特殊情况. 例如, 由于一些 I 帧数据包较大, 超过了 MTU 的最大范围 1 500, 因此需要使用 RTP 协议对 H.264 数据包进行拆包分片, 这种情况下, H.264/AVC 视频码流中就不再存在完整的关键帧, 无法直接对关键帧进行加密, 相反, 需要对已经分片的关键帧进行重新组装, 然后再进行加密. 在第 3.3 节中, 详细介绍了使用国产 SM2 算法对获取的关键帧 NALU Header 数据进行加密的方法, 同时也涉及到了加密过程中所使用的秘钥信息. 最后, 在第 3.4 节中, 以 MP4 文件为例, 详细说明了视频媒体文件加密处理的流程以及核心算法.

3.1 视频帧类型判断

为了实现本文对视频媒体文件加密效率的目的, 首先需要对视频帧类型进行判断并找到关键帧, 在前面已经提到, 每个 NALU 由一个字节的 Header 和 RBSP 组成, 其中 NALU Header 由 3 部分组成, forbidden_bit (1 bit), nal_ref_idc (2 bits) 代表优先级, nal_unit_type (5 bits) 代表 NALU 的类型. 本文通过表 1 所示计算方法

可以判断出 NALU 中数据的载荷类型, 计算 Header & 0x1F 的值, 若值等于 1, 便进而找到了视频帧序列中的 I Frame. 由于 P Frame 和 B Frame 计算方式相同, 所以此方法无法区别出 P Frame 和 B Frame, 但不影响我们对于 I Frame 的判断.

表 1 NALU 载荷类型判断表

Nal_unit_type	计算方法
SPS: 0x67	Header & 0x1F = 7
PPS: 0x68	Header & 0x1F = 8
SEI: 0x66	Header & 0x1F = 6
I Frame: 0x65	Header & 0x1F = 5
P Frame: 0x41	Header & 0x1F = 1

图 3 表示使用 Elecard Stream Analyzer 工具随机截取的一组 H.264/AVC 码流信息. 通过图 3 可以直观地观察到 I、P、B 帧的位置关系, 码流起始部分包含了 SEI、SPS、PPS、I Slice 这 4 个结构, 总长度为 0x15AC7, 后面的 P Slice 以及 B Slice 就各自单独作为一帧, 第 1 个 P 帧长度为 0x10D36, 第 1 个 B 帧长度为 0xA5D8, 由此可知我们通过表 1 的计算方法找到视频序列中所有 I 帧数据, 对 I 帧数据的部分信息进行加密处理后, 进而会影响到整个 H.264/AVC 码流信息, 之后再将码流数据封装成各种视频媒体文件时导致视频文件无法正常播放, 达到加密视频媒体文件的效果.

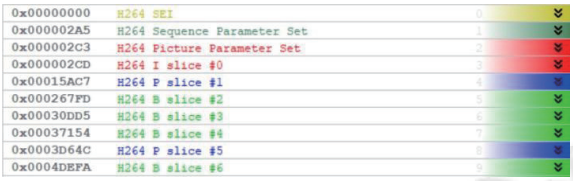


图 3 H.264/AVC 码流示意图

3.2 MP4 文件加密方法研究

本文的视频数据信息通过使用 RTSP 协议向前端摄像头请求视频流获得, 前端摄像头将 H.264 裸码流打包后进行网络传输, 接收端接收后进行组包还原裸码流, 在网络传输的过程中, 一些情况下由于一些 I 帧数据包比较大, 已经超过了 MTU 的最大范围 1 500, 所以就需要拆包分片传输, 这里说的拆包发送不是指发送超过 1 500 的数据包时 TCP 的分段传输或者 UDP 的 IP 分片传输, 而是指 RTP 协议本身对 H.264 的拆包发送与接收, 在视频流传输过程中 H.264 的 RTP 打包有如下 3 种方式.

(1) 单 NALU: P 帧或者 B 帧比较小的数据包, 直接将 NALU 打包成 RTP 包进行传输 RTP Header (12 B) + NALU Header (1 B) + NALU Payload.

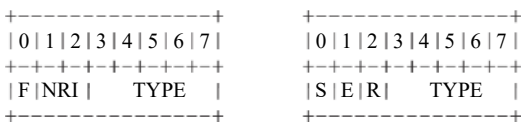
(2) 多 NALU: 特别小的数据包直接将几个 NALU 放在一个 RTP 包中.

(3) FUs (fragment units): 一般情况下, 当 I 帧长度超过 MTU 的范围, 就必须要拆包组成 RTP 包了, 拆分成 FU-A 和 FU-B 两种 RTP 包, 而 RTP 打包方式就变成了 RTP Header (12 B) + FU Indicator (1 B) + FU Header (1 B) + NALU payload.

本节研究的第 3 种打包方式中实际上 NALU 头部信息被分散填充到 FU Indicator 和 FU Header 里面了, bit 位按照从左到右编号 0–7 来算, NALU 头中 0–2 前 3 个比特位放在 FU Indicator 的 0–2 前 3 个比特位上, 后 3–7 这 5 个比特位放入 FU Header 的后 3–7 这 5 个比特位中, 因此查看 I 帧 P 帧类型, 遇到 FU 分片的, 直接看第 2 个字节, 即 FU-B 后 5 位, 这个跟直接看 NALU 头并无差异. 通过式 (1) 计算出 NALU Header 原始数据信息.

$$\text{NALU Header} = (\text{FU Indicator} \,\&\, 0\text{xE0}) \,|\, (\text{FU Header} \,\&\, 0\text{x1F}) \tag{1}$$

如图 4 中 FUs 结构图所示, 在 FU Indicator 结构中若 TYPE 值为 28 代表 FU-A 分片, 若 TYPE 值为 29 则代表 FU-B 分片. 在 FU Header 结构中 1 bit 的 S 表示开始位, 与 E 所表示的结束位是互斥关系, 若 S 位置为 1 则表示 NALU 单元分片后跟随的 NALU payload 是 NALU 载荷的开始部分, 代表一个 I 帧分片的开始, 否则表示一个 I 帧分片的结束, 本文通过找到 I 帧分片开始的数据包, 最终还原出 NALU Header 的原始数据并使用国密算法对得到的原始数据进行加密处理.



(a) FU Indicator (b) FU Header

图 4 FUs 结构图

3.3 国密 SM2 加密处理

国产 SM2 算法是我国自主研发的公钥密码算法, 基于椭圆曲线多倍点运算单向函数, 现行使用的秘钥长度为 256 位, 加密算法选择国产 SM2 算法而不使用国际加密算法的原因是利用非对称加密算法所提供的高安全性的同时可以有效避免使用国际密码技术带来

的安全风险, 从而确保加密过程中的安全性.

本文使用开源的 Gmlib 库进行开发, 具体的加密处理过程为, 用户首先提供一个用于加密的字符串 da_hex="3945208F_7B2144B1_3F36E38A_C6D39F95_88939369_2860B51A_42FB81EF_4DF7C5B8", 然后使用函数 bint_from_str(& da, da_hex, 16) 生成用户私钥 BINT da, 拥有私钥信息后我们就可以通过 ec_mul(&P, &da, &SM2_Fp256_CTX.G, &SM2_Fp 256_CTX) 函数来构造椭圆多倍点 (仿射坐标), 同时生成公钥信息 ECPoint P, 此时我们就可以使用公钥信息并利用 sm2_encrypt(out, &outl, msg, sizeof(msg), PC, &SM2_Fp256_CTX, &P) 函数来进行加密, 其中 PC 表示公钥是非压缩表示, msg 是经过使用第 3.1 节和第 3.2 节中所提出的方法获取的 I 帧 NALU Header 数据, 通过 sm2_encrypt 函数来对 msg 数据进行加密, 随后, 将加密后的数据与 H.264 码流一起封装到 MP4 文件中, 由于 NALU Header 数据被加密, 导致无法正确解析 I、P、B 帧, 从而使得 MP4 文件无法正常播放, 这样就实现了使用国密 SM2 加密 MP4 文件的目的, 同时, 与传统加密算法相比, 只需对 H.264/AVC 中的少部分数据进行加密, 大大减少了计算量, 进而提高了视频媒体文件加密的效率. 使用 SM2 加密处理 I 帧 NALU Header 后的效果如图 5 所示.

```
msg:
dump data: size = 2
0000 - 7c 61                                        |a

ciphertext:
dump data: size = 99
0000 - 04 20 02 b0 bb 3c 3d 90 ca 1f 83 06 5a 7e e4 7e    . . . .<=. . . . .Z~.~
0010 - 39 c0 07 c4 75 f5 1b bd a0 1d fe b1 12 71 e6 05    9. . .u. . . . . .q. .
0020 - f4 6e ed 65 d9 2b ec 38 9d 4b 38 49 5f 0c 5f 62    . n .e .+.8 .K8I _. _b
0030 - 75 17 09 19 dd ae 20 35 a2 06 a5 5b 53 95 41 f9    u . . . . . 5. . . [S . A.
0040 - 26 13 e5 40 e8 72 1e 08 e5 90 62 9f be 36 80 d4    &. .@ . r. . . .b. .6. .
0050 - c3 b3 c7 8b 8e a0 a7 7c 86 53 7f a2 ba 80 97 a6    . . . . . |. S . . . . . .
0060 - c1 f7 d0                                           . . .

msg(decrypt):
dump data: size = 2
0000 - 7c 61                                        |a
```

图 5 SM2 加密 I 帧 NALU Header 效果图

在图 5 中 msg 表示包含 NALU Header 信息的两字节数据"|a", 经过国密 SM2 加密后生成图中所示的密文文件 ciphertext, 此时加密工作已经完成, MP4 文件已经无法打开. 对加密后视频媒体文件解密方式的思路与加密方式相反, 主要是对 I 帧的部分数据加密后, 再无法区别 I 帧, 所以采用将所有其他载荷类型 SPS、PPS、P Frame、B Frame 等都排除后就只剩下加密后的 I 帧或 non-IDR 帧, 然后就可以进行解密处理了, 而本文只是在加密数据包后直接使用用户私钥 BINT da 信息, 通过 sm2_decrypt(out, &outl, out, outl, &SM2_Fp256_CTX, &da) 函数进行解密处理, 得到图 5 中的 msg(decrypt) 解密数据信息"|a"与加密前的原始数据内容一致, 最终成功解密出加密内容.

3.4 MP4 文件加密处理流程及算法

通过以上研究内容我们使用国密 SM2 算法来对我们计算出来的 I 帧 NALU Header 信息进行加密, 导致 I 帧、P 帧、B 帧都无法被正确解析, 成功加密 MP4 文件, 这种加密方法避免了传统加密方法中需要加密所有视频帧数据的繁琐过程, 并减少了加密时间, 在实现有效加密 MP4 文件的同时提升了加密的效率, 图 6 为加密 MP4 文件方法的流程图.

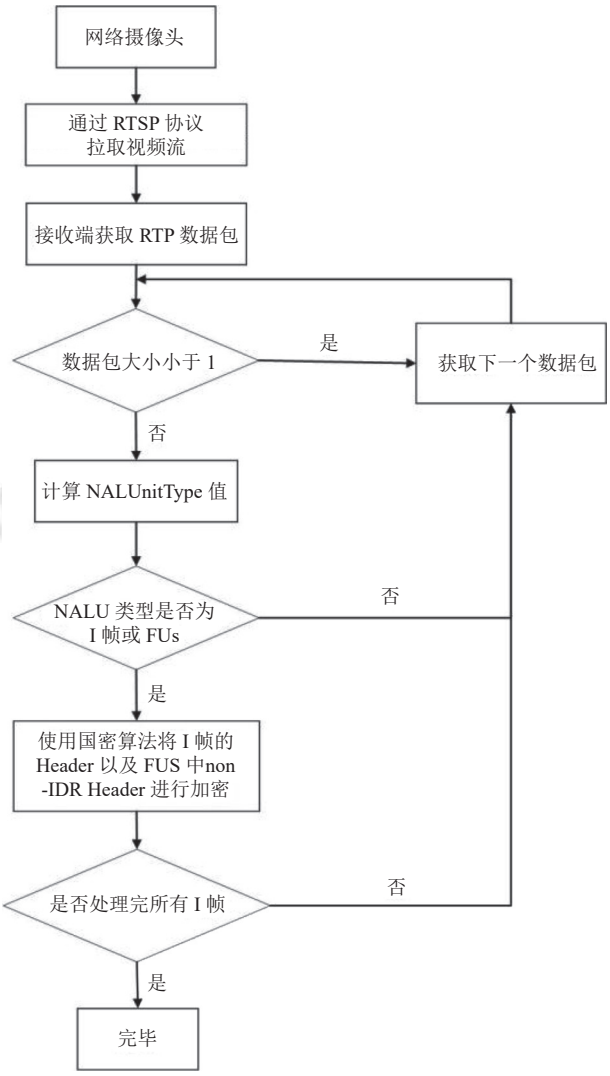


图 6 加密 MP4 文件方法流程图

有效加密 MP4 文件的方法的具体实现步骤如下.

(1) 首先通过 RTSP 协议 rtsp://admin@10.10.3.178:554/h264/ch01/main/av_stream 访问海康网络摄像头拉取视频编码格式为 H.264 视频流信息.

(2) 客户端接收到包含网络摄像头视频流信息的 RTP 数据包.

(3) 客户端对接收到的 dateSize 进行判断, 若 dataSize ⩾ 1则计算出 NALUnitType 的值. 若dataSize < 1 则读取下一个数据包, 进入到步骤 (2) 中.

(4) 对计算出来的 NALUnitType 的值通过表 1 所示的计算方法以及图 4 中 FUs 类型判断方式, 判断出 NALU 类型是否为 I 帧或者是 FUs, 若是其中之一则进行下一步. 否则读取下一个数据包, 进入到步骤 (2) 中.

(5) 将步骤 (4) 中得到的 I 帧头部信息或者是 FUs 中还原出来的 NALU Header 或 non-IDR Header 数据使用国密 SM2 算法对其进行加密处理.

(6) 最后判断是否处理完所有 I 帧数据信息, 若全部处理完成则加密 MP4 文件的流程完毕. 否则读取下一个数据包, 进入到步骤 (2) 中.

为了提升视频媒体文件的加密效率, 本文设计了如下算法, 通过该算法可以清楚地看出本文对视频媒体文件进行加密处理的方式.

算法 1. 视频媒体文件加密效率提升算法

输入: 媒体数据包 headerStart, 媒体数据包大小 dataSize.
输出: 媒体数据包密文 ciphertextPacket.

1. 如果packetSize<1, 媒体数据包不含关键帧, 算法结束.
2. 计算headerStart[0]&0x1F, 得到当前媒体数据包中 NALU 的类型值.
3. 判断当前的 NALU 类型.
4. 若 NALU 类型值等于 5, 表示媒体数据包中包含的是关键帧, 加密步骤: 通过椭圆曲线函数构造 SM2 算法, 生成私钥 BINT Da 和公钥 ECPoint P 信息, 将关键帧的 NALU Header 数据使用 sm2_encrypt 和公钥 ECPoint P 进行加密, 得到密文数据, 使用 NALU Header 密文数据替换原始媒体数据包 headerStart 中 NALU Header 信息, 算法返回 ciphertextPacket.
5. 若 NALU 类型值等于 28 或 29, 表示媒体数据包中包含的是分片的关键帧或 non-IDR 信息, 使用本文中式 (1) 还原出 NALU Header 原始数据信息, 然后执行步骤 4 中的加密步骤.
6. 若 NALU 类型值不等于步骤 4 或步骤 5 中的值, 算法结束.

4 实验结果与分析

为了验证本文提出的基于国密算法的视频媒体文件加密效率提升技术的有效性, 使用国密 SM2 算法对 H.264/AVC 视频帧中 I 帧 NALU Header 数据进行加密处理的方法, 对 MP4 文件做加密处理, 加密后的 MP4 文件在当前市面上大多数的多媒体播放软件上进行测试, 均无法正常播放, 加密效果比较理想, 可以对 MP4 文件的内容起到很好的保护作用. 为进一步证明加密方法的安全性, 本节将分别进行视觉加密效果分析、峰值信噪比测试、结构相似度测试和抗统计攻击测试, 并通过使用对比实验的方式来对视频媒体文件的加密效率进行分析, 最后对视频加密方法的实际应用效果进行评估.

4.1 实验背景与平台

操作系统: Ubuntu 20.04 LTS
处理器: Intel Core i5-8300H CPU@ 2.30 GHz
网络摄像头: 海康威视 DS-2CD3232(D)-I5
流媒体服务器: Live555 Media Server
开发语言: C/C++语言

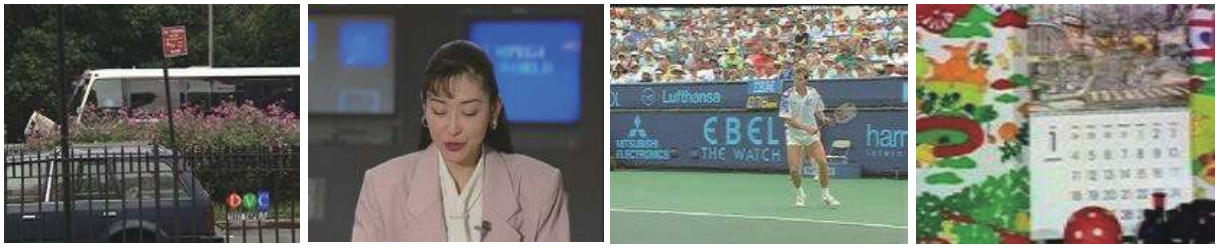
4.2 加密算法的安全性和效率分析

为评估视频加密算法的有效性及安全性, 本节使用 H.264 标准编码器 JM 18.6 对加密算法的安全性和效率进行测试和分析. 为使结果具有代表性, 从标准视频库中选择了 4 个视频序列进行分析, 其中包括两个 CIF 格式的视频 (Bus, Presenter) 以及两个 QCIF 格式的视频 (Tennis, Mobile). 每个视频序列都展示出了不同的场景组合, 如快速运动、复杂纹理和静止背景, 其中 Bus 序列呈现了摄像机的移动拍摄效果, Presenter 序列包含静态背景和活动前景, Tennis 序列中有快速移动的目标, 而 Mobile 序列则具有复杂的纹理和运动信息.

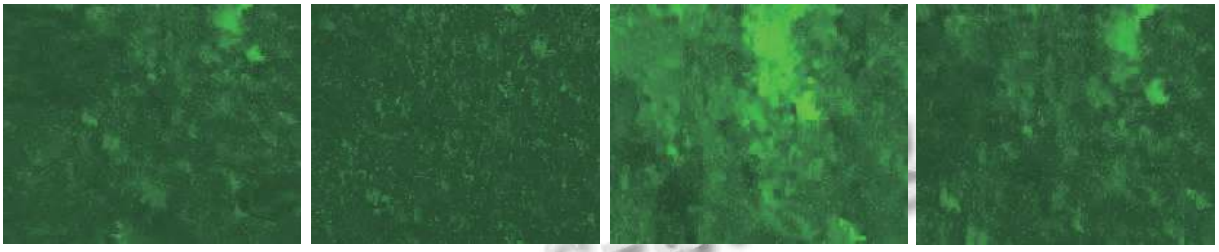
4.2.1 视觉加密效果分析

随着视频媒体文件的生成, 加密过程也同步进行, H.264/AVC 视频序列中关键帧的 NALU Header 信息在此期间被置乱和替代, 使得视频图像的颜色、纹理细节都发生了严重扭曲和模糊变形, 导致视频内容无法理解, 从而达到加密视频的目的. 这里使用了 4 个基准视频序列来测试加密性能, 视频加密前和加密后的效果如图 7 所示, 图 7(a)–图 7(d) 是加密前的图像效果, 图 7(e)–图 7(h) 是加密后的效果.

从图 7 的对比效果中可以很清楚地发现, 加密后的视频图像与原始图像相比, 无论是颜色信息还是纹理信息都发生了巨大的变化, 加密后的视频内容被严重扭曲和模糊, 且没有明文块泄露, 视频内容无法被人眼理解和识别, 这表明视频加密算法满足了视觉安全性的要求.



(a) Bus 加密前 (b) Presenter 加密前 (c) Tennis 加密前 (d) Mobile 加密前



(e) Bus 加密后 (f) Presenter 加密后 (g) Tennis 加密后 (h) Mobile 加密后

图 7 视觉加密效果测试

4.2.2 峰值信噪比和结构相似度测试

除了在主观视觉上证明了视频加密算法的安全性, 我们还需要采用 PSNR (峰值信噪比)、SSIM (结构相似性指数) 这些客观评价指标来衡量视频的感知质量. PSNR 被广泛用于客观地评估视频质量, 通常认为, 当图像的 PSNR 值低于 20 dB 时, 图像内容就无法被识别. 与 PSNR 不同, SSIM 是一种更适合人类视觉系统主观感知的评价方法, 它通过测量图像结构失真程度来评估视频质量, SSIM 的取值范围为 0–1, 数值越接近 1 则说明视频图像质量越高. 表 2 给出了 4 个测试视频序列前 50 帧加密前后的平均 PSNR 值和加密后的 SSIM 值, 并将其作为评估指标.

表 2 不同测试序列的 PSNR 和 SSIM 加密前后对比

视频序列	类型	PSNR (dB)		SSIM
		原始序列	加密序列	
Bus	CIF	36.47	10.38	0.22
Presenter	CIF	37.75	13.64	0.36
Tennis	QCIF	35.67	7.63	0.17
Mobile	QCIF	34.32	9.98	0.21

表 2 中的结果显示, 所有类型的原始序列视频在未加密时的 PSNR 值均超过 40 dB, 这表明它们具有良好的视频质量和高清晰度. 然而, 在加密后所有视频的 PSNR 值均低于 11 dB, 这意味着加密后的视频质量严重降低, 已难以辨识. 而加密后的视频 SSIM 值远小于 1, 趋近于 0, 这表明加密后的视频质量与原始视频质量相比有很大差距. 同时, 相比于 QCIF 格式的视频, 可以看出加密操作对 CIF 格式的视频影响更大, 其加密后的 SSIM 值更低, 加密效果更明显. 因此本文中的加密算法在 PSNR 和 SSIM 的客观评价上也是安全的.

4.2.3 抗统计攻击测试

统计分析攻击是一种通过对加密数据的统计特征和模式进行分析, 以推断出原始数据或密钥信息的攻击方法. 然而, 在本文的选择性加密过程中, 为确保加密算法的实时性和可操作性, 加密数据只是编码过程中的语法元素. 此外, 多媒体数据不仅是简单的二进制数据, 而是复杂的组织结构, 因此, 攻击者无法仅通过统计分析攻击来获取所有明文或秘钥信息. 为便于分析, 从 Bus 视频序列中随机提取了一张明文图像和对应的加密图像, 并绘制了如图 8 所示的统计直方图, 视频帧的直方图反映了像素频率的分布. 对于一个良好的加密算法, 原始视频和加密视频的视频帧直方图要有着明显差异, 而直方图之间的差异越大, 则说明加密算法的加密效果越好, 因此, 视频帧直方图是衡量加密算法效果优劣的重要指标之一.

通过观察图 8 可以发现, 明文图像的直方图与密文图像的直方图具有较大差异, 没有明显的统计规律, 这一结果表明, 本文提出的视频加密方法可以有效地抵抗统计攻击.

4.2.4 视频媒体文件加密效率分析

为了充分分析本文提出的加密方法是否满足对视频媒体文件加密效率提升的要求, 将通过对比实验进行分析, 实验中使用本文提出的加密方法和传统视频

加密方法中的非对称 RSA 加密算法[22]来分别对 100–500 MB 的 MP4 文件 (由 5 个标准视频序列转换而来) 进行加密, 本文的加密方法主要针对 H.264/AVC 码流中关键帧的 NALU Header 数据进行加密, 而 RSA 算法则将整个 MP4 文件作为一个二进制文件, 并直接对全文件按字节进行加密. 对比实验的数据在表 3 中记录, 表 3 中的加密所需时间表示使用 SM2 算法和 RSA 算法对同一个 MP4 文件进行加密所花费的时间对比, 而视频文件大小则表示使用 SM2 加密后, 相较于原始文件, 文件大小的变化情况.

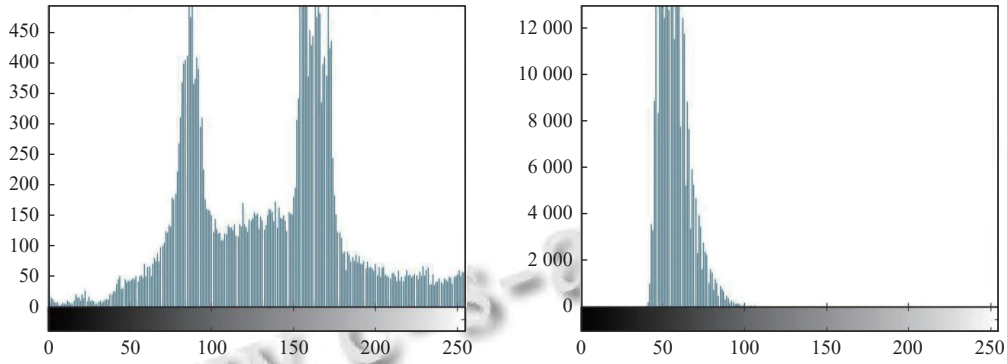


图 8 Bus 视频序列统计直方图测试 (其中, 横坐标表示数据的取值范围, 纵坐标表示数据值出现频率 (frequency))

表 3 加密时间和比特率分析

视频序列	类型	加密所需时间 (s)		视频文件大小 (KB)	
		SM2加密	RSA加密	加密前	加密后
Bus	CIF	36.148	94.264	101 856	101 852
Presenter	CIF	75.399	189.142	205 392	205 388
Tennis	QCIF	120.725	283.265	300 364	300 368
Mobile	QCIF	158.214	364.121	408 908	408 904
Foreman	QCIF	198.265	444.276	492 468	492 464

从表 3 中的数据可以得出结论, 基于 SM2 算法的加密方法相较于 RSA 算法, 在加密同一个 MP4 文件时所需的时间显著减少, 而视频文件的大小在加密前后变化不大, 其影响可以忽略不计, 这意味着加解密前后视频的编码数据量和输入数据量不变, 保持了比特率和压缩比的一致性, 这样做的好处是, 在传输和存储视频数据时不会占用过多的空间, 并且不会增加计算的复杂度和对压缩性能造成影响. 为了更加直观地展示本文提出的加密方法对视频媒体文件加密效率的提升效果, 根据表 3 中的数据绘制出图 9 中的折线图, 来比较 SM2 和 RSA 加密时间之间的差异.

通过分析图 9 可以清楚地得出结论, 本文所使用的加密方法与传统的 RSA 算法相比, 在加密同一个 MP4 文件时, 随着文件大小增大, 加密文件所需时间明显减少, 加密效率显著提升. 图 10 展示了使用 SM2 加密方法只处理包含关键帧的数据包与传统 RSA 加密算法加密所有数据包之间的数量关系.

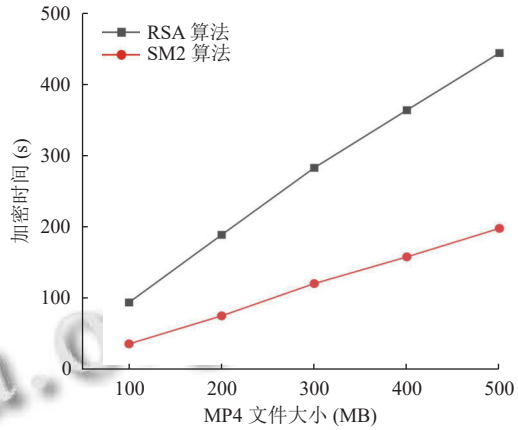


图 9 SM2 和 RSA 加密时间对比折线图

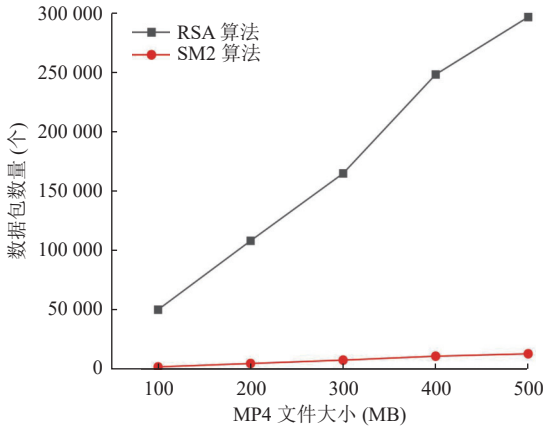


图 10 SM2 和 AES 加密数据包数量折线图

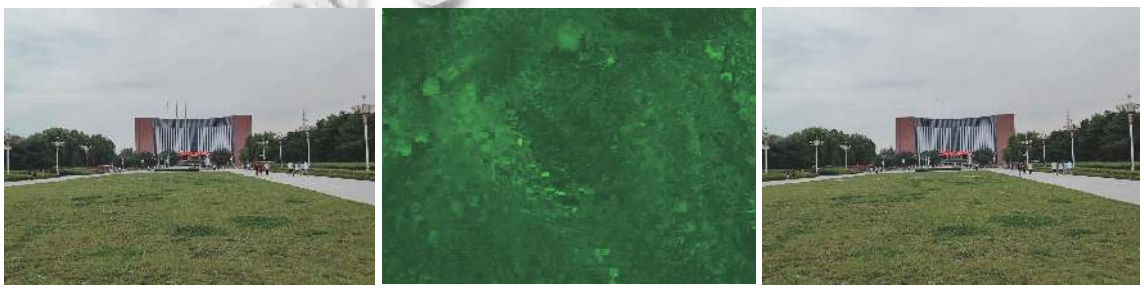
通过观察图 10, 可以明显地发现随着 MP4 文件大小的增加, 本文所使用的加密方法处理的数据包个数明显少于 AES 加密算法处理的数据包数量, 这导致需要进行加密处理的数据量也急剧减少, 最终实现了视频媒体文件加密效率的提升.

通过上述分析, 可以得出以下结论, 相较于传统的视频加密算法, 基于 SM2 算法的加密方法不仅具有更高的加密效率, 而且能够在保障视频数据安全的同时保持比特率和压缩比的一致性, 这一优势对于视频的传输和存储非常有益. 因此, 基于 SM2 算法的加密方法在视频保护领域具有广阔的应用前景, 并为视频内容的安全传输和高效存储提供了技术支持.

4.3 视频加密算法实际应用效果分析

本文介绍了一种针对 H.264/AVC 码流的视频媒体文件加密效率提升技术, 由于 H.264/AVC 在视频编码领域有着广泛的应用和市场, 所以本文提出的加密技术可以在视频监控、视频电视会议、远程医疗和视频保密通信等多个领域中得到应用. 为了检验该加密算法在实际应用中的加密效果, 本文选取了一段大学校园内的监控视频, 并提取其中的 H.264 码流进行加密操作, 其加密效果如图 11 所示.

通过观察图 11, 我们可以看到经过加密后的视频监控码流被杂乱的色块所掩盖, 导致图像无法被人识别和理解, 从而达到了加密的目的, 与此同时, 解密后的视频图像与原图像完全一致, 这证明了该加密算法是无损和可逆的. 由此可见, 该加密算法能够有效地保护视频内容的安全性, 同时在解密过程中不会损失任何信息, 确保了原始图像的完整性和准确性.



(a) 原始图片 (b) 加密图片 (c) 解密图片

图 11 视频码流加密效果测试

通过上述分析可以得出结论, 本文提出的针对 H.264/AVC 码流的视频媒体文件加密效率提升技术在实际的视频监控系统中展现了良好的加密效果, 并且解密后的图像质量也较高, 验证了该技术的有效性. 此外, 该技术的应用领域也可以扩展到所有采用 H.264/AVC 压缩标准的视频流.

5 结语

本文的主要贡献是提出一种基于国密算法对视频媒体文件加密效率提升的技术, 通过对 H.264/AVC 视频帧数据中关键帧的 NALU Header 以及 non-IDR Header 数据信息进行加密的处理方法, 在实现对视频媒体文件的有效加密的同时, 大幅度地提升了加密的效率. 从实验结果分析中可以看出, 该方法加密视频媒体文件性能高效, 相比于传统加密过程中所使用的 RSA 加密算法, 该方法可以有效地减少加密的时间, 减少加密帧的个数、减少加密数据数量, 从而既保证了加密文件的安全性又提高了加密的效率, 能够满足对视频媒体文件加密效率提升的要求.

参考文献

1 Kannan M, Priya C, Vaishnavi Sree S. A comparative analysis of DES, AES and RSA crypt algorithms for network security in cloud computing. Journal of Emerging Technologies and Innovative Research, 2019, 6(3): 574–582.

2 王乐. 互联网背景下数字媒体版权保护的系统设计. 现代电子技术, 2021, 44(10): 143–147. [doi: 10.16652/j.issn.1004-373x.2021.10.032]

3 汪朝晖, 张振峰. SM2 椭圆曲线公钥密码算法综述. 信息安全研究, 2016, 2(11): 972–982.

4 Tabash FK, Izharuddin M, Tabash MI. Encryption techniques for H.264/AVC videos: A literature review. Journal of Information Security and Applications, 2019, 45: 20–34. [doi: 10.1016/j.jisa.2019.01.001]

5 周怡钊, 王晓东, 章联军, 等. 基于 Logistic 和 Arnold 变换的 HEVC 选择性加密方案. 计算机应用, 2019, 39(10):

2973–2979. [doi: 10.11772/j.issn.1001-9081.2019040742]
6 Zeng B, Yeung SKA, Zhu SY, *et al*. Perceptual encryption of H.264 videos: Embedding sign-flips into the integer-based transforms. IEEE Transactions on Information Forensics and Security, 2014, 9(2): 309–320.
7 Khlif N, Damak T, Kammoun F, *et al*. A very efficient encryption scheme for the H.264/AVC CODEC adopted in intra prediction mode. Proceedings of the 2014 International Image Processing, Applications and Systems Conference. Sfax: IEEE, 2014. 1–7.
8 Wang YS, O'Neill M, Kurugollu F. Partial encryption by randomized Zig-Zag scanning for video encoding. Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS). Beijing: IEEE, 2013. 229–232.
9 Ahn J, Shim HJ, Jeon B, *et al*. Digital video scrambling method using intra prediction mode. Proceedings of the 5th Pacific Rim Conference on Advances in Multimedia Information Processing (PCM 2004). Tokyo: Springer, 2005. 386–393.
10 梁剑, 何军辉. 基于宏块编码信息自适应置换的 H.264/AVC 视频加密方法. 计算机科学, 2022, 49(1): 314–320. [doi: 10.11896/jsjkx.201100089]
11 Xu H, Tong XJ, Zhang M, *et al*. Dynamic video encryption algorithm for H.264/AVC based on a spatiotemporal chaos system. Journal of the Optical Society of America A, 2016, 33(6): 1166–1174.
12 韦丞婧, 李国东. 结合超混沌系统和 Logistic 映射的视频图像加密算法. 计算机工程, 2022, 48(5): 263–271. [doi: 10.19678/j.issn.1000-3428.0061608]
13 Sbiaa F, Kotel S, Zeghid M, *et al*. A selective encryption scheme with multiple security levels for the H.264/AVC video coding standard. Proceedings of the 2016 IEEE International Conference on Computer & Information Technology (CIT). Nadi: IEEE, 2016. 391–398. [doi: 10.1109/CIT.2016.53]
14 Shahid Z, Chaumont M, Puech W. Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21(5): 565–576.
15 Chung Y, Lee S, Jeon T, *et al*. Fast video encryption using the H.264 error propagation property for smart mobile devices. Sensors, 2015, 15(4): 7953–7968. [doi: 10.3390/s150407953]
16 Pastuszak G, Trochimiuk M. Architecture design of the high-throughput compensator and interpolator for the H.265/HEVC encoder. Journal of Real-time Image Processing, 2016, 11(4): 663–673. [doi: 10.1007/s11554-014-0422-1]
17 徐辉. 基于混沌的视频数据安全技术研究[博士学位论文]. 哈尔滨: 哈尔滨工业大学, 2019.
18 van Wallendael G, De Cock J, van Leuven S, *et al*. Format-compliant encryption techniques for high efficiency video coding. Proceedings of the 2013 IEEE International Conference on Image Processing. Melbourne: IEEE, 2013. 4583–4587. [doi: 10.1109/ICIP.2013.6738944]
19 Yang MX, Zhuo L, Zhang J, *et al*. An efficient format compliant video encryption scheme for HEVC bitstream. Proceedings of the 2015 IEEE International Conference on Progress in Informatics and Computing. Nanjing: IEEE, 2015. 374–378.
20 Zhu YL, Li XB, Chen N. An algorithm of extracting I-Frame in compressed video. MATEC Web of Conferences, 2015, 22: 01010. [doi: 10.1051/matecconf/20152201010]
21 Zhao LN, Guan LL. An optimized method and implementation for parsing MP4 metadata. Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing. Shanghai: IEEE, 2010. 984–987. [doi: 10.1109/PIC.2010.5687866]
22 李云飞, 刘菊琨, 柳青. 改进 RSA 算法的安全性分析. 计算机应用与软件, 2018, 35(6): 309–312. [doi: 10.3969/j.issn.1000-386x.2018.06.056]

(校对责编: 牛欣悦)