第36卷 第20期 计 算 机 工 程 2010年10月
Vol.36 No.20 Computer Engineering October 2010
·安全技术· 文章编号：1000—3428(2010)20—0145—02 文献标识码：A 中图分类号：TP309.2

格式兼容的视频加密算法设计与实现

朱庆生，刘金凤，葛 亮，赖师悦，罗大江
(重庆大学计算机学院，重庆 400044)

摘 要：现有格式兼容的加密算法在安全性、可操作性和解码耗时等方面各有优劣。针对上述情况，提出一种新的视频加密算法。该算法根据MPEG视频标准，通过改变宏块条竖直位置值进行置乱，实现图像的加密。理论分析与仿真实验结果证明，该算法简单易行，安全性高，加密数据量少，加密速度快，且对数据压缩率无影响。

关键词：格式兼容；宏块条竖直位置；一一映射加密

Design and Implementation of Format-compliant Video Encryption Algorithm

ZHU Qing-sheng, LIU Jin-feng, GE Liang, LAI Shi-yue, LUO Da-jiang
(College of Computer Science, Chongqing University, Chongqing 400044, China)

【Abstract】 Format-compliant video encryption algorithms have their own merits and defects on safety, operability and decoding time-consuming. Aiming at the situation, this paper proposes a novel format-compliant video encryption algorithm which is convenient, secure, fast and has no effect on the compression ratio. Videos are encrypted by permutation through changing the value of the vertical position for macro-blocks. Simulation results on two standard video sequences show that compared with the original one, the new algorithm is more effective.

【Key words】 format-compliant; macro-block-vertical position; one-to-one mapping encryption

1 概述

在提高多媒体加密速度的同时如何保持多媒体的格式兼容开始受到研究者的关注。格式兼容即加密后的多媒体语法不受破坏，仍然能用多媒体播放器播放，只是播放内容会根据安全性要求的不同有不同程度的改变。格式兼容使加密后的文件继承了加密前文件的一些良好特性，如纠错能力、网络友好性、视频的随机访问、自适应带宽、视频转换。格式兼容也使一系列信号处理技术能运用到加密后的文件上。

现有格式兼容的加密算法主要包括：

(1)Zig-zag 置乱算法[1]。用一个随机的置乱序列置乱DCT系数的Zig-zag扫描顺序，从而达到加密视频图像的效果。该算法被认为不够安全，且显著增加了视频码流大小，减小了视频编码的压缩比。

(2)使用分组密码算法对DCT系数或运动矢量符号位进行加密的RVEA算法[2]。它对每个宏块限定最多加密64个符号位。加密符号位的方法计算量小，但仅加密上述符号位不够安全。实验表明，对于加密了上述全部符号位的视频码流，如果将帧内块DC符号位全部设置为某个常量(如128)，那么即使其他DCT和运动矢量符号位不解密，由于DC含有像素块的主要能量信息，因此仍然可以看出图像的大致轮廓。

文献[3]提出一种基于MPEG-4视频的格式兼容加密算法。该算法主要针对VLC编码的代码字。对于一个有N个($N=2^n$)代码字的VLC代码字表来说，需要为其中每个代码字各分配一个n bit的索引号。加密代码字C对应的索引号I得到I'，并用I'对应的代码字C'替换C。得到的加密比特流与原流拥有相同的代码字个数，并能保证加密后的代码字串是有效的。

解码是视频加密过程中非常耗时的一个环节，文献[3]提出的加密算法对保持MPEG-4视频的格式兼容非常有效，但其格式兼容针对的是DC、AC、MV等的VLC编码，不但加密数据量较大，且这些系数处在视频的block层内。采用此加密算法就意味着必须完全解码视频，十分耗时。

本文结合MPEG视频标准和保持格式兼容的思想，提出一种通过改变宏块条竖直位置值的视频加密算法。

2 与算法有关的MPEG视频流格式[4]

MPEG视频流共分为6层：图像序列层，图像组层，图像层，宏块条层，宏块层，块层。视频中定义了3类帧：I帧，P帧，B帧。I帧是帧内编码图像，这类图像不参考其他帧而只利用自己的图像信息进行编码；P帧需要前向预测；B帧需要双向预测。运动预测以宏块为基本单位。因此，P、B存在可跳过宏块，即在解码器中既没有运动向量又没有DCT系数信息的宏块。

根据视频流语义属性，宏块条竖直位置是由32位宏块条层起始符的后8位确定的(当图像很大，竖直方向的像素行大于2 800时，还由占3位的宏块条竖直位置扩展符(slice_vertical_position_ extension)决定，本文不考虑将宏块条竖直位置扩展符也加密)。由于一个宏块条中的第1个和最后一个

基金项目：国家“863”计划基金资助项目“虚拟植物可视化引擎关键技术研究”(2006AA10Z233)；国家自然科学基金资助项目“生理引擎驱动的虚拟植物多分辨率展示关键技术研究”(60773082)

作者简介：朱庆生(1957－)，男，教授、博士生导师，主研方向：图像加密；刘金凤，硕士研究生；葛 亮，讲师、博士研究生；赖师悦、罗大江，硕士研究生

收稿日期：2010-05-23 **E-mail：**ljfcqu@126.com

宏块必须具有同样的垂直位置，因此宏块条是不跨行的。同一个宏块条竖直位置可存在多个宏块条。

3 本文的格式兼容的视频加密算法

3.1 算法基本思想

宏块条的竖直位置由宏块条层起始符的后 8 位指示，可以通过加密宏块条竖直位置值(即加密宏块条层起始符的后 8 位)来加密图像。由于宏块条的竖直位置值是有范围的，因此设 *mbRows* 表示一个视频的一帧中所包含的宏块行数。根据 MPEG 视频格式标准，第 1 行宏块条的竖直位置为 1，所以，宏块条竖直位置值的取值范围为[1, *mbRows*]，其加密后得到的值必须在此范围内。本文同样用 8 位表示加密后的值，从而保证格式兼容性。

对于 I 帧来说，由于它属于帧内编码，没有可跳过宏块，也没有运动预测，因此它的宏块条竖直位置值是连续的，加密宏块条的竖直位置能使图像在竖直方向上出现置乱效果。对 I 帧的宏块条竖直位置加密的效果如图 1 所示。

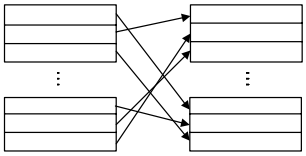


图 1　对 I 帧的宏块条竖直位置加密的效果

对于 P 帧、B 帧来说，除了使竖直位置上出现置乱效果外，由于存在运动预测，因此假设存在运动预测的宏块 *M* 所在的宏块条的竖直位置为 *X*，加密后其值变为 *X'*，对应新宏块位置的宏块为 *M'*。原本需要运动预测的宏块 *M* 没有得到运动补偿，反而对 *M'*进行了运动补偿，运动补偿的错位加强了加密效果。主要加密流程如图 2 所示，缓冲区大小最大为 1 帧数据。

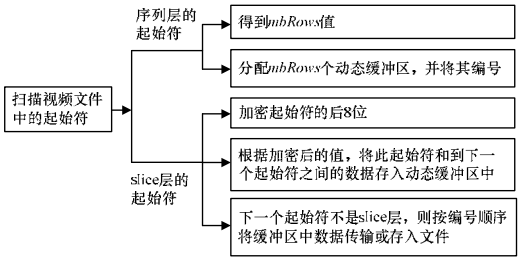


图 2　主要加密流程

3.2 一一映射加密

由于宏块条的竖直位置值是有范围的，因此其加密后得到的值也必须在此范围内。为了保证加密后的效果，应设计一一映射的加密方法。

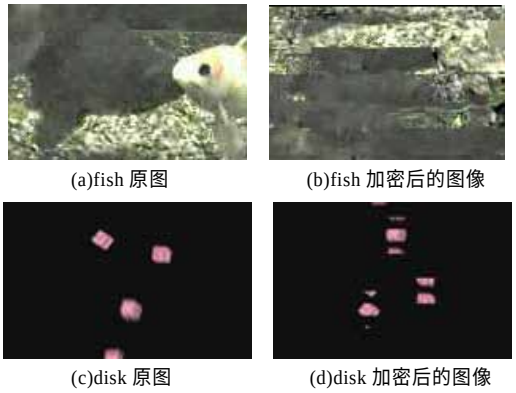
对于每个在取值范围[1, *mbRows*]内的竖直位置值，其加密后的值都能唯一地在该区间内找到，对于每个在[1, *mbRows*]内的加密后的竖直位置值都能唯一地恢复出在[1, *mbRows*]内的原始竖直位置值。可以看出，一一映射加密实际上是得到一种 1~*mbRows* 的整数随机排列。

为达到一一映射加密的目的，本文推荐文献[5]中的 2 种算法。算法 1 将一个 1~*n*!的伪随机整数映射成一个随机排列。该算法的优点是只需要调用一次伪随机数产生器。如果随机数产生函数需要耗费大量计算时间，则使用该算法可以节省很多时间。但是该算法的内存操作时间复杂度为 $O(n^2)$，因此，当 *n* 较大时花费的时间较多。另一方面，当 *n* 较大时，除法和求模运算还需要额外的程序来实现。因此，该算法适合在 *n* 较小时采用。算法 2 对 *n*-1 次交换产生伪随机排列，它调用伪随机数器的时间复杂度为 $O(n)$，内存操作的时间复杂度也为 $O(n)$，因此，该算法总体性能优于算法 1。如果伪随机产生器本身并不耗时，建议使用算法 2 来产生伪随机排列。

为保证算法安全性，本文为每个图像层产生一组随机排列。

4 仿真实验结果

本文选取 2 个具有代表性的视频 fish.mpg 和 disk.mpg 为素材进行仿真实验。图 3 给出了加密前后的图像对比。表 1 描述了文件大小和需要加密的数据量。



(a)fish 原图　(b)fish 加密后的图像

(c)disk 原图　(d)disk 加密后的图像

图 3　加密前后的图像对比

表 1　2 个视频文件的 DC 系数量及生成的位置文件大小

文件名	文件总大小/KB	图像规格	DC 系数量/KB	宏块竖直位置值数据量/KB
Fish	92.2	120×160	1.448	0.712
Disk	36.1	240×320	1.207	0.405

5 算法性能分析

5.1 安全性分析

本文加算密法的安全性主要依赖密钥的安全性，假设一个视频图像包含 *mbRows* 个宏块行，共有 *n* 帧。若采用暴力攻击，由于加密是一一映射加密，因此解码 1 帧需要尝试 *N*=*mbRows*!次。因为本文为每个图像层产生一组随机排列，所以要解码出整个视频需要 N^n 次尝试。考虑一种非常简单的情况，当 *mbRows*=20、*n*=200、$N=(20!)^{20}>2^{100}$ 时，认为该算法是非常安全的。

5.2 加密数据量

由 MPEG 视频标准可知，该算法的数据加密量主要与视频中帧的个数、帧的组成结构和图像竖直方向的宏块数有关。

由表 1 可知，Fish 文件的宏块竖直位置值占文件大小的 0.772%，Disk 文件的宏块竖直位置值占文件大小的 1.1%，可见，本算法加密的数据量很少。

文献[3]算法加密的是 DC、AC、MV 等的 VLC 编码。考察 DC 系数，由表 1 可知 Fish 文件的 DC 系数占文件大小的 1.570%，Disk 文件的 DC 系数占文件大小的 3.343%，DC 系数的平均数据量是宏块竖直位置值的 2 倍以上。因此，本文算法的数据加密量至多是文献[3]算法的 50%。若考虑加上加密 AC、MV 的 VLC 编码，则本算法的优势更明显。

5.3 加密速度

本算法的加密时间主要由 3 个部分构成：扫描文件，产生伪随机排列，读写缓冲区。扫描文件的时间复杂度为 $O(n)$，其中，*n* 由视频文件大小决定。伪随机排列产生的时间复杂度为 $O(n)$，其中，*n* 由帧的宏块行数决定。读写缓冲区的时间复杂度为 $O(n)$，其中，*n* 为缓冲区大小，最大值为 1 帧数据。可见，本算法的总体加密速度很快。　(下转第 149 页)