一种基于多数字基整数的数字水印分存算法

钱言玉 吴友情
(合肥师范学院公共计算机教学部 安徽 合肥 230061)

摘 要 针对一般水印算法功能单一,而现有水印分存算法中只能实现二进制水印图像分存的问题,将多数字基整数应用到数字水印分存技术中,提出一种灰度水印图像的分存方法。实验结果表明,该方法可以有效防止外部欺诈和内部欺诈,实现了在多个用户间的秘密共享,也可以应用于身份认证,且可以抵抗多种常用图像的处理攻击,具有一定的鲁棒性。

关键词 水印分存 多数字基整数 灰度水印 秘密共享 身份认证

中图分类号 TP391.41 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2016.11.072

A WATERMARKING SHARING SCHEME BASED ON MULTIPLE-BASED NUMBER

Qian Yanyu Wu Youqing
(*Department of Public Computer Teaching, Hefei Normal University, Hefei 230061, Anhui, China*)

Abstract Aiming at the problems that general watermarking algorithms only have single function, while existing watermark sharing algorithms can only realise binary watermark image sharing, we proposed a gray-level watermarking image sharing scheme, which applies the multiple-based number to digital watermarking sharing technology. Experimental results showed that the proposed scheme can effectively prevent external and internal frauds, and achieve the secret sharing among multiple users, it can be also applied to authentication, can resist a couple of processing attacks on common images, and has certain robustness.

Keywords Watermarking sharing Multiple-based number Gray-scale watermark Secret sharing Identity authentication

0 引 言

在网络技术的发展下,国内外学者提出了很多数字水印算法,但是这些算法中能解决数字水印分存的问题的很少。因为在这些算法中只有一个密钥,任何人只要他拥有这个密钥,就可以重构数字水印,就会被默认为是作品的拥有者。如果多人协同合作拥有一个数字水印,可以运用某种方法设计出 n 份不同的子水印,分配给 n 位参与者。当获取其中的 $t(t_0 \leqslant t \leqslant n, t_0$ 是指定秘密共享方案固有的阈值)份子水印,共享的数字水印就可以被恢复;反之则无法恢复出共享的数字水印。这就是秘密共享的问题。

数字水印分存是在数字水印的基础上,将秘密共享[1-5]的思想用于数字水印技术中。姚惠明、牛少彰等人[6,7]提出的数字水印分存技术将分存后的子水印嵌入在 LSB 上,只能抵抗剪切攻击,不能抵抗其他类型攻击,健壮性差。基于矢量共享方案 DCT 域上的数字水印分存算法[8]提出在 DCT 变换域中嵌入分存子水印,这在一定程度上增强了子水印的健壮性,但对剪切和压缩之外的攻击效果仍很不理想。基于视觉的秘密共享方案[9]不需要复杂的密码机制和数学运算,可以由人眼直接解码,但这种方案在秘密共享的过程中容易产生噪声。基于图像的多用户共享的数字水印系统[10],把秘密嵌入载体图像,没有产生明显的噪声,但仍只能实现二进制图像的秘密共享。

本文提出的基于多数字基整数的数字水印分存算法,可以实现灰度数字水印图像的共享[11]。多数字基整数[12]用来将灰度数字水印分裂成 n 个没有意义的子水印,分派给 n 个用户再分别嵌入到载体图像的不同区域。本算法可用来实现在多个用户间的秘密共享,也可以应用于身份认证[13,14],且可以抵抗多种常用图像的处理攻击,具有一定的鲁棒性。

1 多数字基整数

定义 1 称整数 $d_{n-1}d_{n-2}\cdots d_0$ 为单数字基整数,如果此整数的所有系数 $d_i(0 \leqslant i \leqslant n-1)$ 均以十进制数 a 为基底,其中 d_i 的取值为 0 到 $a-1$ 的十进制整数。则 $d_{n-1}d_{n-2}\cdots d_0$ 转化为十进制整数为:

$$d_{n-1}d_{n-2}\cdots d_0 = d_0 + \cdots + d_{n-2} \times a^{n-2} + d_{n-1} \times a^{n-1} \quad (1)$$

定义 2 称整数 $d_{n-1}d_{n-2}\cdots d_0$ 为多数字基整数,d_i 为此整数的第 i 位系数,如果每位系数 d_i 以十进制常数 $b_i > 0(i = 0,1,\cdots,n-1)$ 为基底,其中 d_i 的取值范围是 0 到 $b_i - 1$ 的十进制整数。多数字基数 $d_{n-1}d_{n-2}\cdots d_0$ 同样可以表示成 $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}\cdots d_{2(b_2)}d_{1(b_1)}d_{0(b_0)}$,它的十进制结果可以按如下公式进行计算:

收稿日期:2015-07-18。**钱言玉**,实验师,主研领域:信息安全。**吴友情**,讲师。

$$\begin{aligned} & d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}\cdots d_{2(b_2)}d_{1(b_1)}d_{0(b_0)} \\ &= d_{n-1}\times(b_{n-2}\times b_{n-3}\times\cdots\times b_0)+ \\ & \quad d_{n-2}\times(b_{n-3}\times b_{n-4}\times\cdots\times b_0)+\cdots+ \\ & \quad d_2\times(b_1\times b_0)+d_1\times b_0+d_0 \\ &= ((\cdots((d_{n-1}\times b_{n-2}+d_{n-2})\times b_{n-3}+d_{n-3})\times \\ & \quad \cdots+d_2)\times b_1+d_1)\times b_0+d_0 \end{aligned} \tag{2}$$

定义 3 称以十进制常数 $b_{n-1}b_{n-2}\cdots b_0$ 为基底，由定义 2 定义的多数字基整数的集合为多数字基整数系统。

性质 1 任何一个十进制整数都可以转化成一个多数字基整数 $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}\cdots d_{0(b_0)}$，其逆命题也真。

性质 2 一个 n 位的多数字基数 $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}\cdots d_{0(b_0)}$，可以表示 m bit 二值信息，如果其基底 $b_{n-1},b_{n-2},\cdots,b_0$ 满足：

$$m\leqslant\left\lfloor\log_2\left(\prod_{i=0}^{n-1}b_i\right)\right\rfloor \tag{3}$$

2 基于多数字基整数的数字水印分存算法

将灰度数字水印图像通过多数字基整数分成 $n(n\geqslant 2)$ 份子水印后，需要将这些子水印嵌入到载体图像中，这里的水印嵌入与提取算法不限定，只要适合就行。本文采用在小波域上结合人眼视觉特性的灰度水印自适应嵌入与提取算法[15]，将数字水印分成 2($n=2$)份。分派给 2($n=2$)个用户再分别嵌入到载体图像的不同区域，基于多数字基整数的数字水印分存算法实现了灰度数字水印图像的分存，提高了水印的嵌入容量，且与以前的算法相比，具有更强的健壮性和鲁棒性。

2.1 灰度数字水印的分块

我们用多数字基整数将原始数字水印分成若干份，这里 SI 表示原始灰度水印，分开后的子水印用 $PI_0,PI_1,\cdots,PI_{n-1}$ 表示，SI 和 $PI_0,PI_1,\cdots,PI_{n-1}$ 可以定义如下：

$$SI=\{s(i,j)\mid 0\leqslant s(i,j)\leqslant 255\quad 0\leqslant i,j\leqslant N-1\} \tag{4}$$

$$PI_k=\{p_k(i,j)\mid 0\leqslant p_k(i,j)\leqslant d_k\quad 0\leqslant i,j\leqslant N-1\} \tag{5}$$

$PI_k(k=0,1,\cdots,n-1)$ 的大小同 SI，$d_k=b_k-1$，b_k 为对应用户所取的整数基底。若取两位基底为 $b_1=18$，$b_0=22$，则整数 200 可以被如下系数表示 $d_1=9$，$d_0=2$，即 $200=d_1\times b_0+d_0=9\times 22+2$。

如图 1 所示，$PI_i(i=0,1)$ 是原始数字水印 SI 分块后的 2 块子水印，它们是分别对应系数 $d_i(i=0,1)$ 的矩阵。这样我们就可以将这些分块的矩阵，也就是子水印嵌入到载体图像的不同区域中。

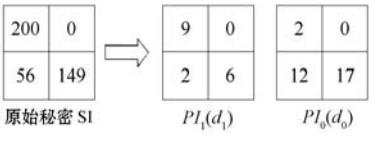

图 1 多数字基整数的数字转换

2.2 子水印的嵌入与提取

本算法的流程如图 2 所示，其主要步骤如下：

步骤 1 为了增强安全性，在原始灰度数字水印图像分裂前，采用基于队列变换的数字图像置乱算法[16]对其进行置乱预处理。基于队列置乱变换有四个参数：$s\_key(Type,I,J,L)$，其中 $Type$ 表示置乱变换类型，(I,J) 表示置乱变换的参照点，L 表示置乱变换迭代的次数。

步骤 2 用多数字基整数将置乱后的灰度数字水印图像分裂成 $n(n\geqslant 2)$ 份子水印图像。相应的多数字基基底可以作为密钥，被对应的用户所拥有。

步骤 3 将各子水印图像嵌入载体图像的不同区域中。

步骤 4 提取各子水印图像，原始数字水印图像可以按照定义 2 进行恢复后再进行逆置乱获得。

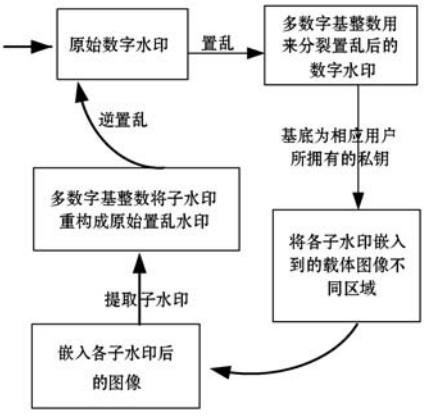

图 2 基于多数字基整数的数字水印分存算法

3 实验结果与分析

为测试本文算法的有效性，在仿真实验中，水印采用 256 级的大小为 64×64 的灰度图像。为了提高水印的安全性和抗剪切攻击能力，采用队列变换的置乱算法先对原始水印进行预处理，再采用合适的基底将置乱后的数字水印表达为两组不同的多数字基整数，即两块子水印。这里的载体图像采用 256 级灰度，大小为 512×512 的“Lena”、“Goldhill”、“Camera”、“Airplane”和“Peppers”图像分别试验。限于篇幅，本文以 Lena 图像为代表，将其进行三级小波变换，两块子水印分别嵌入到对应的载体图像的低频域中和中频域中。原始水印、置乱后的水印如图 3 所示。

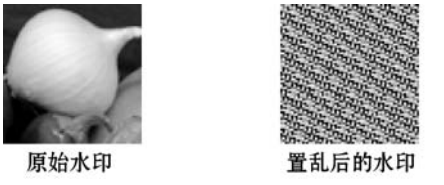

图 3 原始水印和置乱水印

图 4 为原始载体图像，图 5 为嵌入 2 个子水印后的带水印图像，从视觉效果看，很难感觉到水印的存在，说明该水印算法具有较好的不可见性。

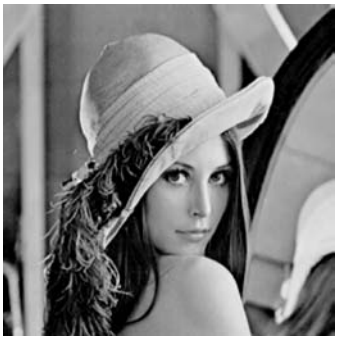

图 4 原始载体图像

图 5　嵌入两块子水印后的图像 $PSNR = 34.5578$

3.1　鲁棒性与安全性分析

为验证基于多数字基整数的数字水印分存算法的抵抗攻击能力,实验中对嵌入水印后的图像进行一些处理,然后提取子水印。归一化互相关系数 NC 来判断提取水印的鲁棒性,用来定量评价提取水印 W_{extract} 与原始水印 W 两者之间的相似度[15]。

$$NC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[(w[i][j]-A)\times(\hat{w}[i][j]-\hat{A})]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(w[i][j]-A)^2}\times\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(\hat{w}[i][j]-\hat{A})^2}} \tag{4}$$

式中,$A = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N}w[i][j]}{M\times N}$,$\hat{A} = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N}\hat{w}[i][j]}{M\times N}$,$M\times N$ 为图像尺寸,$w[i][j]$ 与 $\hat{w}[i][j]$ 分别表示原始水印 W 与提取水印 W_{extract} 中第 i 行和第 j 列处的像素值。A 和 $\hat{A}$ 分别表示 W 与 W_{extract} 的平均像素值。

如表 1 所示,“子水印 1 + 子水印 2”表示两个子水印都被提取,“子水印 1”表示只有第一块子水印被提取,第二块子水印的密钥未知。“子水印 2”表示只有第二块子水印被提取,第一块子水印的密钥未知。为了分析基于多数字基整数的数字水印分存算法的安全性,我们从外部欺诈和内部欺诈两个方面来考虑:就外部欺诈而言,敌手很难获得 2 份子水印,再完成逆置乱,从而恢复原始数字水印;就内部欺诈来说,内部子水印合法拥有者在不知道其他子水印的情况下,即使提取出了自己的子水印,也变得毫无价值。所以本文提出的数字水印分存算法,不仅能防止外部欺诈,也能防止内部欺诈,达到了秘密共享的目的。

表 1　水印抗攻击实验结果

对含水印图像的攻击方法	子水印 1 + 子水印 2	子水印 1	子水印 2
缩放	0.9169	0.1931	0.0451
高斯滤波	0.6454	0.1411	0.0384
JPEG 压缩 $q=50$	0.8823	0.1851	0.0424
JPEG 压缩 $q=46$	0.8758	0.1807	0.0402
JPEG 压缩 $q=42$	0.8739	0.1724	0.0332
JPEG 压缩 $q=40$	0.8647	0.1703	0.0359
去除位平面 0	0.9426	0.2006	0.0397
去除位平面 0、1	0.9382	0.1996	0.0375
去除位平面 0、1、2	0.9141	0.1947	0.0317
椒盐噪声(0.01)	0.6120	0.1458	0.0349
乘性噪声(0.01)	0.6506	0.1300	0.0103
旋转 (0.1)	0.7089	0.1533	0.0412

3.2　实验结果比较及分析

运用数字水印技术来分享一个秘密,同种类型的算法并不多。一般说来,不容易在这类算法中来比较彼此性能的优劣,因为它们使用不同的规则用不同的方法来达到秘密共享的目的。在表 2 中我们列出了几条一般性能的比较。

表 2　与相关系统的性能比较

基准	视觉秘码学	一般的数字水印系统	一般的基于视觉的水印系统	文献[10]	本文算法
鲁棒性	×	○	○	○	○
不可见性	×	○	○	○	○
秘密分享	○	×	×	○	○
嵌入容量	×	○	○	○	○
安全性	○	○	○	○	○
计算复杂度低	○	×	×	×	○

对于数字水印系统,像不可见性和鲁棒性是考虑的重点,但秘密共享能力一般并不考虑。视觉密码学主要考虑的是秘密共享能力,为了达到这个目的,恢复出的秘密的质量欠考虑,恢复的效果往往也不理想。基于多数字基整数的数字水印分存算法能实现秘密的共享,同时能克服上述缺陷。表 2 中“○”表示“重要”或“拥有”,“×”表示“不考虑”或“不重要”。

4　结　语

数字水印分存是将秘密共享的思想引入数字水印技术中,秘密共享在重要信息和秘密数据的安全保存、传输及合法利用中起着关键的作用。共享方案在使用的时候,一方面可以防止外部欺诈获得秘密;另一方面可以防止内部欺诈,内部子秘密合法拥有者在不知道其他分存水印的情况下,即使提取出了自己的分存水印,也变得毫无价值。这在现实中是非常有意义的,可以防备子秘密合法拥有者盗窃秘密。本文提出了一种能简单有效实现灰度数字水印图像分存的算法,利用多数字基整数将原始数字水印分裂成多个子水印,当且仅当获得 $t(t_0 \leqslant t \leqslant n, t_0$ 为阈值)份子水印时(本文中 $t=2, t_0=2$),原始数字水印才可以恢复。实验表明,基于多数字基整数的数字水印分存算法能有效抵抗外部欺诈和内部欺诈,同时实验结果表明,该算法具有很强的鲁棒性和不可见性,实现了灰度数字水印图像的多用户共享。

参　考　文　献

[1] 刘海,彭长根,田有亮,等. (2,2)贝叶斯理性秘密共享方案[J]. 电子学报,2014,42(12):2481 - 2488.

[2] Zhang Zhifang, Liu Mulan. Rational secret sharing as extensive game [J]. Scinece China Information Sciences, 2013, 56(3): 1 - 13.

[3] 沈刚,郁滨. 基于异或的(k,n)多秘密视觉密码[J]. 小型微型计算机系统,2013,34(9):2116 - 2119.

[4] Harn L, Fuyou M, Chang C C. Verifiable secret sharing based on the Chinese remainder theorem [J]. Security and Communication Networks, 2014, 7(6): 950 - 957.

[5] 蓝才会,王彩芬. 一个新的基于秘密共享的条件代理重加密方案[J]. 计算机学报,2013,36(4):895 - 902.

[6] 姚惠明,隋爱芬,牛少彰,等. 基于矢量共享方案的数字水印分存算法[J]. 电子与信息学报,2003,25(12): 1612 - 1616.

[7] 牛少彰,钮心忻,杨义先,等. 基于拉格朗日插值公式的数字水印分存算法[J]. 北京邮电大学学报,2003,26(3):8 - 11.

[8] 姚惠明,周冠玲,杨义先,等. 一种基于矢量共享方案的 DCT 域上数字水印分存算法[J]. 计算机学报,2004,27(7):998 - 1003.

[9] Fuhgwo Jeng, Kaisiang Lin, Chihhung Lin, et al. Visual multi-secret sharing with friendliness [J]. Journal of Shanghai Jiaotong University (Science), 2014, 19(4): 455 - 465.
[10] Wang F H, Yen K K, Jain L C, et al. Multiuser-based shadow watermark extraction system [J]. Information Sciences, 2007, 177(12): 2522 - 2532.
[11] 罗斌, 吴友情, 吕皖丽, 等. 一种基于相位相关拼接的数字水印分存算法[J]. 中国科技论文, 2009, 4(2): 103 - 108.
[12] Wu D C, Tsai W H. Data hiding in images via multiple-based number conversion and lossy compression [J]. IEEE Transactions on Consumer Electronics, 1998, 44(4): 1406 - 1412.
[13] 周晓斌, 许勇, 张凌. 一种开放式 PKI 身份认证模型的研究[J]. 国防科技大学学报, 2013, 35(1): 169 - 174.
[14] 刘云芳, 左为平. 基于身份无可信中心的指定验证人代理多重签名方案[J]. 计算机应用与软件, 2013, 30(4): 316 - 318.
[15] 向德生, 熊岳山, 朱更明. 基于视觉特性的灰度水印自适应嵌入与提取算法[J]. 中国图象图形学报, 2006, 11(7): 1026 - 1035.
[16] 李敏, 费耀平. 基于队列变换的数字图像置乱算法[J]. 计算机工程, 2005, 31(1): 148 - 152.

(上接第 300 页)

名,通过这种方法可以在一定程度上防止反编译,但是又不影响 apk 包的正常安装。

3 实现与评价

在 Android 应用市场(http://www.appchina.com/,截至 2015 年 4 月)随机选取游戏类、资讯阅读类、系统工具类、生活实用类等 5 款类型不同且大小也不同的 apk(目标 APK 未加壳)进行加固方案测试,加固前后的应用大小变化如表 2 所示。

表 2 加固后应用包大小变化(单位:字节)

应用	加固前	加固后	增加大小	增加比例
com.diota.android.smswishes_44.apk	950 724	1 310 193	359 469	37.8%
com.baowa.wirelessdisk.apk	2 550 106	3 043 725	493 619	19.3%
58daojiaguest.apk	6 941 608	7 492 177	550 569	7.9%
com.droidhen.turbo.apk	16 198 700	16 873 737	675 037	4.2%

测试结果表明:经过加固后应用程序大致增加了 400 ~ 500 KB的容量。加壳后 APK 变大的主要增量是 libsecurity.so 动态库和解壳工程。

通过对常见的逆向工具测试,加固方案效果如表 3 所示。

表 3 加固方案对抗逆向工具评测

加固技术 逆向工具	反编译,静态逆向分析			动态逆向分析	
	apktool	dex2jar	JEB	IDA pro	NetBeans
AXML 插入非法 ID	√	X	X	X	X
加壳	√	√	√	X	X
Ptrace 等反调试	X	X	X	√	√
签名校验	X	X	X	X	√
加固方法综合	√	√	√	√	√

注:√表示加固方法对某逆向工具有效,X 表示加固方法对该逆向工具无效

测试结果表明:不同的加固方法对不同的逆向工具有效。通过 apktool 漏洞可以使得 apktool 反编译失败;加壳可以使得 apktool、dex2jar 或 JEB 无法反编译出真正的 dex;通过 ptrace 检测/proc 文件夹等反调试措施可以防止 IDA Pro、NetBeans 的动态调试;签名校验可以防止 NetBeans 动态调试。综合上述,所有加固方法可以对目标 APK 起到有效全面的保护。

4 结 语

通过对 Android 安全机制的研究,分析常见逆向攻击机制,设计了一种移动应用加固方案。该方案将加密后的目标 dex 嵌入图片可以有效隐藏目标 dex,实验表明,加壳保护技术可以有效防止目标 APK 被静态分析。使用 ptrace 和检测/proc/pid/status 和 inotify 监控关键文件,可以有效完成反调试保护。此外,加固方案还加入了 JNI 层的签名校验,能有效防止重打包。加固方案的最后一个阶段还利用 apktool 解析漏洞添加非法 id 值,使得壳 APK 无法被反编译。通过测试,加固后的 APK 文件会大致增加 400 ~ 500 KB 的容量,对 APK 运行或加载效率造成较小影响。本加固方案没有实现对抗 dump 内存的逆向攻击方法,攻击者仍然可以通过内存拷贝等方式,找到目标 dex 代码。

本加固方案仍然有需要改进和完善的地方,在接下来工作中将会着重研究:1) so 加壳保护。dex 加壳无法防止内存 dump,但通过 so 加壳保护则能大大增加内存 dump 的难度。2) AndroidManifest.xml 配置保护。AndroidManifest.xml 配置文件作为应用程序的“说明书”,当前加固方案对其的保护不足,这是另一个研究重点。

参 考 文 献

[1] 工信部. 移动互联网白皮书[R]. 北京, 2014.
[2] 巫志文, 李炜. 基于 Android 平台的软件加固方案的设计与实现[J]. 电信工程技术与标准化, 2015(1): 33 - 37.
[3] 吴善崇, 张权. Android 平台安全机制浅析[J]. 实验科学与技术, 2014, 12(2): 43 - 45.
[4] 徐剑, 武爽, 孙琦, 等. 面向 Android 应用程序的代码保护方法研究[J]. 信息网络安全, 2014(10): 11 - 17.
[5] 梅瑞, 武学礼, 文伟平. 基于 Android 平台的代码保护技术研究[J]. 信息网络安全, 2013(7): 10 - 15.
[6] 伍景珠. 基于 Android 平台的软件保护方案的研究与实现[D]. 北京邮电大学, 2013.
[7] 丰生强. Android 软件安全与逆向分析[M]. 北京: 人民邮电出版社, 2013.
[8] 张志远, 万月亮, 翁越龙, 等. Android 应用逆向分析方法研究[J]. 信息网络安全, 2013(6): 65 - 68.
[9] 刘劼. Java 反编译技术和代码安全[J]. 现代电子技术, 2004, 27(10): 22 - 24.
[10] 姚为光. 软件加壳技术的研究[D]. 电子科技大学, 2011.
[11] 李文. 基于壳技术的软件保护研究[D]. 电子科技大学, 2012.
[12] 李宇翔, 林柏钢. 基于 Android 重打包的应用程序安全策略加固系统设计[J]. 信息网络安全, 2014(1): 43 - 47.