第35卷 第10期
Vol.35 No.10
计 算 机 工 程
Computer Engineering
2009年5月
May 2009
·安全技术·
文章编号：1000—3428(2009)10—0142—03 文献标识码：A 中图分类号：TP301.6

基于汉字拼音声调的文本水印算法

赵 理[1,2]，崔杜武[1]
(1. 西安理工大学计算机科学与工程学院，西安 710048；2. 石家庄职业技术学院计算机系，石家庄 050081)

摘 要： 提出一种基于汉字拼音声调的中文文本水印算法。该算法基于统计特征来动态确定嵌入标志代码。在由标志代码确定的水印插入区，通过改变汉字集合声调的特征值来嵌入文本水印。该方法的水印容量由标志代码的数量动态确定，可自主地提高水印容量。整篇文档可以分割成若干个嵌入部分，各部分可单独进行插入、提取计算，降低了计算的复杂性。

关键词： 文本水印；鲁棒性；水印容量

Text Watermark Algorithm Based on Tone of Chinese Characters

ZHAO Li[1,2], CUI Du-wu[1]
(1. School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048;
2. Department of Computer, Shijiazhuang Vocational Technology Institute, Shijiazhuang 050081)

【Abstract】 A text watermark algorithm based on the tone of Chinese characters is proposed. The text watermark, with the modification of the characters value by transformations, can be embedded into the area by the marked codes. The marked codes are decided by the characteristic of the document. The watermark capacity decided by the amount of marked codes can be changed freely. The whole article can be divided into many parts, the embedding and extraction of the text watermark can be done independently in each. The complexity can be reduced in this way.

【Key words】 text watermark; robustness; watermark capacity

1 概述

当前的文本数字水印主要分为基于格式的文本数字水印和基于自然语言处理的文本数字水印。由于基于格式的文本水印算法完全依赖于文本格式，水印信息是嵌入在文本内容之外的，因此抗攻击性不强、鲁棒性较差[1-4]。而基于自然语言的文本水印算法将水印嵌入在文本内容之中，和基于文本格式的水印算法相比，它的抗攻击性好。具有更好的鲁棒性，是未来文本水印方法的主流方向[5-8]。

但是当前大多数基于自然语言的文本水印算法普遍存在以下缺陷：(1)水印容量普遍不足；(2)适用于中文特点的文本水印方法研究较少；(3)不能同时较好地满足不可见性和鲁棒性的要求。基于以上缺陷，本文提出了一种基于汉字拼音声调变化的文本水印算法，并引入冗余机制，嵌入信息的单位不再局限于句子，大大地提高了嵌入信息的容量，具有一定的抗攻击性。

2 算法的基本思想

汉语的每个文字在特定的语言环境下都有一个唯一确定的汉语拼音声调。这些声调包括：1声，2声，3声，4声。整篇文章可以看作是由这些声调组成的集合。

假定一段文档 M 由 n 个字组成，第 $i(1\leqslant i\leqslant n)$ 个字用 C_i 表示。用 ξ 表示各个汉字的二进制特征值。

设定如下规则：

(1)若 C_i 发音为1声，则 $\xi(C_i)=001$。

(2)若 C_i 发音为2声，则 $\xi(C_i)=010$。

(3)若 C_i 发音为3声，则 $\xi(C_i)=011$。

(4)若 C_i 发音为4声，则 $\xi(C_i)=100$。

$\xi(M)=\xi(C_1)\cup\xi(C_2)\cup\cdots\cup\xi(C_n)$，表示整篇文章由一系列0，1组成的二进制代码组成。

2.1 嵌入水印信息的数量的计算

对要嵌入的信息进行二进制编码，比如要插入的水印信息为“西”。则对应的汉字代码为1100111011110111(西)。对要嵌入的水印前后分别加入起始、结束标志，分别为“00000001”和“00000100”。通过计算可知，需要插入的信息有32位。

2.2 嵌入标志位的选取

从头开始对由 $\xi(M)$ 表示的二进制代码进行扫描。从文章中找到出现次数大于要嵌入的水印信息位数且代码段长度大于等于6的二进制代码段集合(因为一个字符占用3个二进制位，为保证2个标志代码之间的文本至少有2个文字的间隔，所以要求2个代码段间隔6个二进制位以上，重复次数才能记为1次)。从中选取出现次数最少的代码段，作为嵌入信息开始的标志，这样可以保证嵌入的水印信息均匀地分布在文章中。

2.3 嵌入标志位之间的代码序列特征值计算

特征值计算规则如下：

假定一段文档中有 i 个嵌入标志位，将第 i 到第 i+1 个标志位之间的代码定义为 d_i。用 t 表示嵌入标志位之间的代码序列特征值。则：当 d_i 中二进制位值为“1”的个数为奇数时，$t(d_i)=1$；当 d_i 中二进制位值为“1”的个数为偶数时，$t(d_i)=0$。

2.4 改变特征值的方法

通过对标志位间的汉字、词或句子的处理，可以改变标志位间代码的特征值。主要的处理方式包括：句子的主动式和被动式之间的变换，同义词的替换，句子的压缩和扩展等。

作者简介： 赵 理(1974－)，男，讲师、硕士研究生，主研方向：信息安全；崔杜武，教授、博士、博士生导师

收稿日期： 2008-10-14 **E-mail：** zhao_li@sjzpt.edu.cn

2.5 水印的嵌入

顺序将待嵌入的水印信息位与各个标志代码段之间的代码序列的特征值进行比较。通过使用改变特征值的各种方法，对与待嵌入的水印信息位不一致的代码序列的特征值进行变换，使之相符。

2.6 水印的提取

对于文档中水印的提取，其实就是文档中嵌入水印信息的反过程。提取步骤如下：

(1)从头开始对由$\xi(M)$表示的二进制代码进行扫描。找到代码段长度大于等于 6 的所有重复的二进制代码段的集合。这些代码段各自将文章分割成若干个代码区间，从中找到前 8 个区间特征值为“00000001”的二进制代码段集合。如果集合内元素大于 1，则从中找到有连续 8 个区间特征值为“00000100”的二进制代码段，这个代码段就是要找的标志代码段。

(2)从头开始对对标志代码段之间的代码序列求特征值。

(3)对得到的特征值按顺序 8 位一组，进行拼接，连接成嵌入水印的二进制代码。去掉起始、结束标志后，得到水印的原文。

3 实验及性能分析

3.1 实验

笔者对来自网络的一段文章进行了实验，原文如图 1 所示，在图中用不同的大小的汉字表示找到的标志代码段。

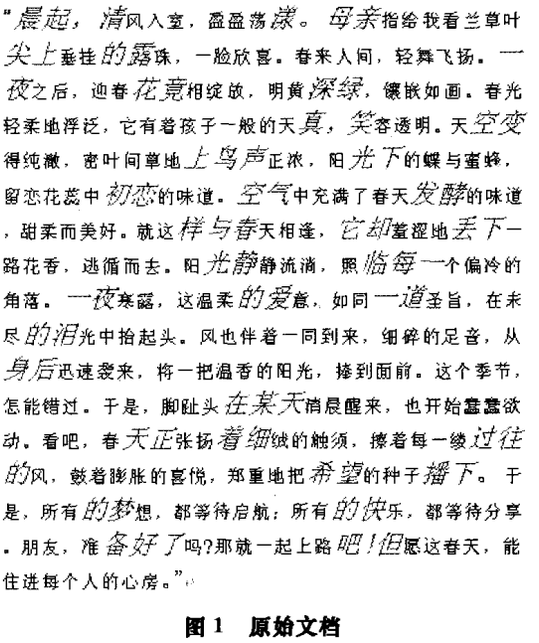
“晨起，清风入室，盈盈荡漾。母亲指给我看兰草叶尖上垂挂的露珠，一脸欣喜。春来人间，轻舞飞扬。一夜之后，迎春花竟相绽放，明黄深绿，镶嵌如画。春光轻柔地浮泛，它有着孩子一般的天真，笑容透明。天空变得纯澈，密叶间草地上鸟声正浓，阳光下的蝶与蜜蜂，留恋花蕊中初恋的味道。空气中充满了春天发酵的味道，甜柔而美好。就这样与春天相逢，它却羞涩地丢下一路花香，逃循而去。阳光静静流淌，照临每一个偏冷的角落。一夜寒露，这温柔的爱意，如同一道圣旨，在未尽的泪光中抬起头。风也伴着一同到来，细碎的足音，从身后迅速袭来，将一把温香的阳光，捧到面前。这个季节，怎能错过。于是，脚趾头在某天清晨醒来，也开始蠢蠢欲动。看吧，春天正张扬着细绒的触须，擦着每一缕过往的风，鼓着膨胀的喜悦，郑重地把希望的种子播下。于是，所有的梦想，都等待启航；所有的快乐，都等待分享。朋友，准备好了吗?那就一起上路吧!但愿这春天，能住进每个人的心房。”

图 1 原始文档

假如要嵌入的水印信息为“西”，它对应的编码为“1100111011110111”，加上起始、结束标志 “00000001”和“00000100”，则要嵌入的二进制代码为“00000001110011101111011100000100”。利用规则 1，计算出由$\xi(M)$表示的上文的二进制代码。从$\xi(M)$的第 1 位二进制代码开始扫描，找到重复次数多于 32 的且位数大于等于 6 的二进制代码段的集合，共有 3 个，分别为“001100(33 次)”、“100100(37 次)”、“001001(36 次)”。在位数为 7 的二进制代码段中，重复次数最大的是“0110010(26 次)”，次数太小，不满足要求。所以，选取“001100”作为嵌入标志代码。在上文中用不同的字体表示出了包含嵌入标志的文字。

对文本进行扫描，用“001100”将原文档代码分隔成若干个区间。对于各区域的特征值，计算结果如下：

$t(d1)=0$, $t(d2)=0$, $t(d3)=0$, $t(d4)=0$, $t(d5)=0$, $t(d6)=1$, $t(d7)=1$, $t(d8)=0$, $t(d9)=0$, $t(d10)=0$, $t(d11)=0$, $t(d12)=1$, $t(d13)=1$, $t(d14)=1$, $t(d15)=0$, $t(d16)=1$, $t(d17)=1$, $t(d18)=0$, $t(d19)=1$, $t(d20)=1$, $t(d21)=1$, $t(d22)=0$, $t(d23)=1$, $t(d24)=1$, $t(d25)=1$, $t(d26)=0$, $t(d27)=0$, $t(d28)=0$, $t(d29)=0$, $t(d30)=0$, $t(d31)=1$, $t(d32)=0$

将水印信息和 $t(di)$进行对比如下：

“00000001 11001110 11110111 00000100”

“00000110 00011101 10111011 10000010”

可知，需要对 $t(d6)$, $t(d7)$, $t(d8)$, $t(d9)$, $t(d10)$, $t(d12)$, $t(d15)$, $t(d16)$, $t(d18)$, $t(d21)$, $t(d22)$, $t(d25)$, $t(d30)$, $t(d31)$进行变换。利用改变特征值的方法，改变后的文章如图 2 所示。

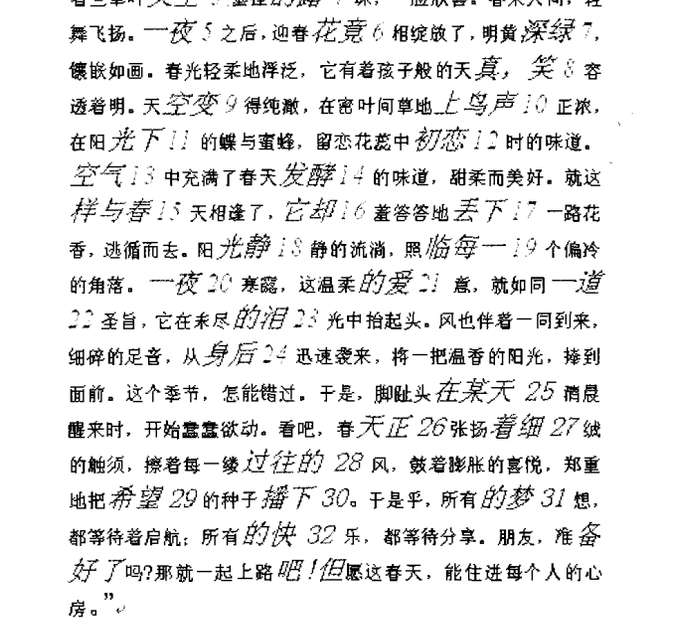
“晨起，清1风入室，盈盈荡漾2。母亲3指给我看兰草叶尖上4垂挂的露5珠，一脸欣喜。春来人间，轻舞飞扬。一夜6之后，迎春花竟7相绽放了，明黄深绿8，镶嵌如画。春光轻柔地浮泛，它有着孩子般的天真，笑9容透着明。天空变10得纯澈，在密叶间草地上鸟声11正浓，在阳光下12的蝶与蜜蜂，留恋花蕊中初恋13时的味道。空气14中充满了春天发酵15的味道，甜柔而美好。就这样与春16天相逢了，它却17羞答答地丢下18一路花香，逃循而去。阳光静19静的流淌，照临每一20个偏冷的角落。一夜21寒露，这温柔的爱22意，就如同一道23圣旨，它在未尽的泪24光中抬起头。风也伴着一同到来，细碎的足音，从身后25迅速袭来，将一把温香的阳光，捧到面前。这个季节，怎能错过。于是，脚趾头在某天26清晨醒来时，开始蠢蠢欲动。看吧，春天正27张扬着细28绒的触须，擦着每一缕过往的29风，鼓着膨胀的喜悦，郑重地把希望30的种子播下31。于是乎，所有的梦32想，都等待着启航；所有的快33乐，都等待分享。朋友，准备好了吗?那就一起上路吧!但愿这春天，能住进每个人的心房。”

图 2 嵌入水印后的文档

上文中对嵌入的区间号用数字做出了标记，在对文章进行变换时要注意，变换后不要在该变换区间产生额外的标志代码段。提取水印时，先在$\xi(M)$中扫描位数为 6 的重复次数大于 16 的二进制代码段。开始、结束标志占用 16 位，所以重复次数最少要大于 16。结果显示共有 23 种。对其中每一种进行计算，看由它做标志段时，前 8 个标志段之间的代码特征值是否为“00000001”。结果显示，只有“001100”符合条件，所以“001100”就是要找的标志段。继续计算标志段之间的代码特征值，直到得到的特征值为“00000100”时停止。就这样找到了嵌入的水印信息，去掉起始、结束标志，得到“西”的二进制代码。

3.2 性能分析

3.2.1 关于水印容量的对比与讨论

在目前常用的基于自然语言的文本水印算法中，基于词性标记串统计特性的文本水印算法[9]水印容量高于本算法，但是嵌入水印后的载体文本容易发生语义改变和难以理解的情况，不可见性不够理想。基于句子长度的中文文本水印算法水印容量低于本算法，但是该算法的计算量也低于本算法。基于汉字笔画统计特征的文本水印算法[10]水印容量也低于本算法。利用“的”字嵌入水印的算法[11]的水印容量和本文中的算法接近，但其鲁棒性不佳。本文提出的算法，水印容量可根据需要，在一定范围内调整。其理论最大值为 $n/4$(n为文档汉字总数)。

3.2.2 关于水印嵌入、提取计算量的探讨

本算法在水印嵌入和提取时的计算量，随着待插入文档长度的增加而急剧放大，严重影响了运算效率。所以，当文档长度过大时，可以先将文档按代码长度分割成若干段，水印的嵌入、提取以这些段为单位。每段中的水印有各自的段的开始、结束标志。总的文档中的水印有总的开始、结束标志。这样极大地减少了运算量。

3.2.3 鲁棒性的探讨

由于本文用二进制标志段将$\xi(M)$分割成若干个代码区间，嵌入的水印信息均匀散布在这些区间内，因此对攻击十分敏感。微小的对加了水印的文档的改变，都有可能影响水印的提取。所以，当文档的长度允许时，加大水印标志代码之间的间隔。水印的嵌入、提取不再以标志代码段之间的代码总长为单位，而是以总长的一部分为单位。这样，当水印标志代码之间的间隔很大，而选取插入水印的区间的长度又相对较小时，可明显地提高水印的鲁棒性，但是降低了水印的容量。为进一步提高水印的抗攻击性，还可以在嵌入水印时使用冗余、增设校验位等方法。

4 结束语

区别于以往的基于自然语言的文本水印算法，本文提出了一种新的算法，该方法较好地保持了数字水印的基本特征，具有以下特点：(1)针对中文汉字特征，采用基于汉语拼音音调的方法。(2)水印插入不再基于句子，采用可变的标志代码段标志嵌入区域，较好地解决了水印容量不足的问题，且水印容量可作柔性调整。(3)采用分段计算、冗余插入、校验位校验的方法，在降低计算量的同时，保证了鲁棒性。(4)水印的提取不需和原文进行对比，提高了提取的可操作性。实验表明，该算法有一定的实践意义。

参考文献

[1] Brassil J, Low S, Maxemchuk N F, et al. Electronic Marking and Identification Techniques to Discourage Document Copying[J]. IEEE Journal on Selected Areas in Communications, 1995, 13(8): 1495-1504.

[2] 邹昕光, 孙圣和. 基于RTF格式的文本脆弱水印算法[J]. 计算机工程, 2007, 33(4): 131-133.

[3] 刘旻昊, 孙堡垒, 郭云彪, 等. 文本数字水印技术研究综述[J]. 东南大学学报, 2007, 37(1): 225-230.

[4] 袁树雄, 孙星明. 英文文本多重数字水印算法设计与实现[J]. 计算机工程, 2006, 32(15): 146-148.

[5] Mikhail A J, Victor R, Mchae C, et al. Natural Language Watermarking Design, Analysis, and a Proof-of-concept Implementation[C]//Proc. of the 4th International Information Hiding Workshop. Pittsburgh, USA: [s. n.], 2001: 185-199.

[6] Mikhail A J, Craig M J, Victor R. Natural Language for Information Assurance and Security: An Overview and Implementation[C]//Proc. of New Security Paradigm Workshop. New York, USA: ACM Press, 2000: 51-65.

[7] Gaurav G, Jose P, Wang Huaxiong. An Attack-localizing Watermarking Scheme for Natural Language Documents[C]//Proc. of the ACM Symposium on Information, Computer and Communications Security. Taipei, China: [s. n.], 2006: 157-165.

[8] 戴祖旭, 洪 帆, 董 洁. 基于Huffman编码的文本信息隐藏算法[J]. 计算机工程, 2007, 33(15): 147-148.

[9] 戴祖旭, 洪 帆, 崔国华, 等. 基于词性标记串统计特性的文本数字水印算法[J]. 通信学报, 2007, 28(4): 108-113.

[10] 辛友强, 刘东苏. 一种基于汉字特征和语义的文本数字水印算法[J]. 计算机应用, 2007, 27(S2): 134-135.

[11] 赵敏之, 孙星明, 向华政, 等. 基于不完整语义理解的文本数字水印算法研究[J]. 计算机应用研究, 2006, 23(4): 118-120.

编辑 顾逸斐

(上接第 141 页)

随机化是没有意义的。也就是说，一次随机化即可达到防御目的。这样，最优的情况是攻击者每次成功猜测后都进行一次随机化。这时攻击者不可能猜中 2 个以上的目标地址。假设只进行一次随机化，则其结果与这次随机化发生的时机有关。如果这次随机化发生在攻击者第 1 次猜测成功之后，攻击者就必须再猜测 T 个目标，总共要猜测 T+1 次。如果这次随机化发生在 T-1 次猜测之后，那么攻击者总共需要猜测的次数是 3T-2 次。假设在猜测 T 个目标的过程中的 T-1 个时机中随机选择 $t(0<t<T)$个时机进行随机化，则要进行的成功猜测次数平均为

$$\frac{1}{C_T^t}\times\left(C_{T-1}^{t-1}\times\frac{T(T-1)}{2}\right)=\frac{t(T-1)}{2}$$

再对 t 取平均值，则平均而言要进行的成功猜测次数为 $\frac{T(T-1)}{4}$，也即将攻击者需要猜中的次数提高到了平方数量级。攻击者成功实施攻击的概率降低为 $\left(\frac{1}{N}\right)^{\frac{T(T-1)}{4}}$。

6 结束语

本文提出的运行时刻地址空间重复随机化技术，以最有效的随机化时机尽可能使攻击者已经获得的信息失效，从而进一步增大攻击成功实施的难度。运行时刻随机化技术在进程运行的任意时刻能随机化其满足一定条件的动态共享库。下一步的研究方向是扩展可被随机化的对象，更好地确定运行时刻再次随机化的时机，并将运行时刻再次随机化技术同其他防御技术相结合。

参考文献

[1] 王立民, 曾凡平, 李 琴. 基于指针备份的随机化技术[J]. 计算机工程, 2007, 33(17): 87-189.

[2] Bhatkar S, Sekar R, DuVarney D C. Efficient Techniques for Comprehensive Protection from Memory Error Exploit[C]//Proc. of of the 14th USENIX Security Symposium. Baltimore, MD, USA: [s. n.], 2005.

[3] Kil C, Jun J, Bookholt C, Xu Jun, et al. Address Space Layout Permutation(ASLP): Towards Fine-grained Randomization of Commodity Software[C]//Proc. of ACSAC'06. Shanghai, China: [s. n.], 2006.

[4] Xu Jun, Kalbarczyk Z, Iyer R K. Transparent Runtime Randomization for Security[C]//Proc. of of the 22nd Symposium on Reliable and Distributed Systems. Florence, Italy: [s. n.], 2003.

[5] Xu Hanzhi, Chapin S J. Improving Address Space Randomization with a Dynamic Offset Randomization Technique[C]//Proc. of the ACM Symposium on Applied Computing. Dijon, France: ACM Press, 2006: 384-391.

[6] Shacham H, Page M, Pfaff B, et al. On the Effectiveness of Address-space Randomization[C]//Proc. of the 11th ACM Conference on Computer and Communications Security. Washington D. C., USA: ACM Press, 2004: 298-307.

编辑 金胡考