第35卷 第20期 计 算 机 工 程 2009年10月
Vol.35 No.20 Computer Engineering October 2009
·安全技术· 文章编号：1000—3428(2009)20—0153—04 文献标识码：A 中图分类号：TP391.1

基于子元素排列组合的XML文档信息隐藏

杨 洁[1]，张秋余[2]，王丽敏[1]，芮雄丽[1]
(1. 南京工程学院通信工程学院，南京 211167；2. 兰州理工大学计算机与通信学院，兰州 730050)

摘 要：分析XML文档的层次结构，提出基于XML子元素排列组合的信息隐藏算法。将待隐藏秘密信息转换成十进制整数，利用子元素的排列组合形成等价元素，根据等价元素与整数间的映射关系，采用等价元素置换方法将整数嵌入XML文档。实验结果和分析表明，该算法不改变XML文件大小，其隐蔽性和鲁棒性优于现有XML文档信息隐藏技术，且信息隐藏量较大，可以应用于XML网页保护和隐秘通信。

关键词：信息隐藏；可扩展标记语言；等价元素；排列组合

XML Document Information Hiding Based on Sub-element Permutation and Combination

YANG Jie[1], ZHANG Qiu-yu[2], WANG Li-min[1], RUI Xiong-li[1]
(1. College of Communication Engineering, Nanjing Institute of Technology, Nanjing 211167;
2. College of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050)

【Abstract】This paper analyzes the hiberarchy of eXtensible Markup Language(XML) document. Based on XML sub-element permutation and combination, an information hiding algorithm is presented. This algorithm changes secret information to a decimal integer and creates equal element by permutation and combination of sub-element. According to mapping relation between equal element and integer, the integer is embedded into the XML document by changing the element to its equal element. Experimental results and analysis show that the algorithm does not change the size of XML document, and has better imperceptibility and robustness and larger information hiding capacity than existing XML document hiding techniques. The algorithm can be used to protect the content of a XML Web page and covert communication.

【Key words】information hiding; eXtensible Markup Language(XML); equal element; permutation and combination

1 概述

信息隐藏[1]是指在图像、视频、音频、文本、网页等载体中嵌入一些秘密信息，从而让第三方在主观上难以察觉秘密信息的存在，主要包括用于隐秘通信的隐写术[2]和用于数字媒体版权保护的数字水印技术[3]。目前，信息隐藏技术已成为信息安全和多媒体版权保护的一个研究热点[4-7]。

文本信息具有信息量大、传递快捷等特点，是人们广泛使用的信息交换载体。目前，Web应用的主要载体是HTML，但随着人们需求的不断提高，Web应用空前复杂，HTML的局限性逐渐显现。W3C开发了一种超越HTML的语言——XML，由于其使用灵活、可扩展性强，因此逐渐取代HTML语言成为互联网上数据交换的标准。

文献[8]提出基于XML句法规则和文档逻辑结构的信息隐藏算法，该算法改变XML文档中标记的属性表示方法和命名，变换标记的字体，通过改变同名元素的排列顺序以及标签之间的尺寸进行信息隐藏。本文将通过改变标记之间的尺寸来隐藏信息的算法记为算法1。

文献[9]提出通过对文件中的元素名称进行同义词替换来实现信息隐藏。先获得XML文档中所有元素节点的元素名称，为每个元素名称构造一组同义词，形成元素名称的同义词库，然后按序遍历文档中的各个元素，根据待隐藏信息的值按序对各元素的名称进行替换，以实现信息隐藏。本文将该算法记为算法2。

文献[10]提出根据XML的编码规则，在使用引号时，双引号和单引号作用相同，因此，可以作为信息隐藏载体，在编码时单引号表示0，双引号表示1。本文将其记为算法3。

针对现有XML文档信息隐藏技术隐蔽性差、隐藏容量小的缺点，本文提出一种基于XML元素排列的信息隐藏算法，利用子元素的排列组合形成等价元素，采用等价元素置换方法隐藏信息。

2 XML文件结构

XML是SGML的优化子集，它是一种元语言。其基本思想是数据按固有结构以层次化格式存储，从而形成一种基于内容的格式。XML与HTML相似，但各方面性能优于HTML。XML使用的标记由用户根据自己的需要定义，极大增强了文件可读性。XML的内容和显示格式相分离，通过使用样式表语言可以很方便地实现同一内容的不同显示。

XML文档中数据包含在元素中。一个元素被定义为一个开始标签、一个结束标签和标签之间的文本。元素可以包含以下内容之一：数据(不是标签部分的文本)，一个或多个其他元素、数据和元素的组合。

基金项目：南京工程学院科研基金资助项目(KXJ08066)；甘肃省自然科学基金资助项目(0803RJZA024)
作者简介：杨 洁(1979－)，女，讲师、硕士，主研方向：网络通信，信息安全；张秋余，副研究员；王丽敏，副教授、硕士；芮雄丽，讲师、硕士
收稿日期：2009-05-13 **E-mail：**yangjie@njit.edu.cn

一个完整的 XML 文档可以映射为一幅文档树图，根节点是文档节点，它代表整个 XML 文档，文档节点有序言节点、注释和文档根元素节点 3 个子节点。序言节点是对 XML 的版本声明、字符编码和文档类型定义等说明信息。文档根元素节点主要描述文件内容，它可以包括元素子节点，元素子节点又有自己的子节点，节点之间通过嵌套表达复杂的文件结构。一个简单的 XML 文档如下：

```
<?xml version="1.0"?>
<people>
  <person>
   <name>JAM</name>
   <telephone>
     <home>55-12</home>
     <office>99-12</office>
     <mobile>123</mobile>
   </telephone>
  </person>
  <person>
    <name>BOB</name>
    <telephone>
      <home>33-18</home>
      <office>77-12</office>
      <mobile>456</mobile>
    </telephone>
  </person>
</people>
```

上述 XML 文档映射的文档树如图 1 所示。

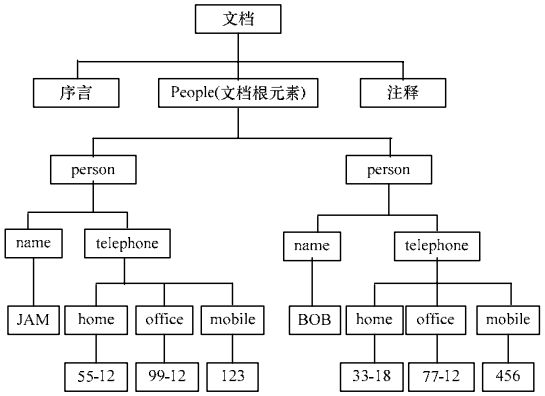

图 1　XML 文档映射的文档树

由图 1 可以看出，people 元素含 2 个子元素，person(左)和 person(右)元素分别含 2 个子元素，telephone(左)和 telephone(右)元素分别含 3 个子元素，对每一级子元素的顺序分别进行排列组合，可以得到 2!×2!×2!×3!×3!=288 种子元素排列，每种排列行成的元素都是等价的，若每种等价元素代表一种信息，则通过等价元素置换可以表示 288 种信息。

3　相关定义

定义 1 令 $E(a_1,a_2,\cdots,a_n)$ 表示 XML 中包含 2 个或 2 个以上子元素的元素(文档元素除外)，其中，E 表示元素的名称；n 表示 E 的子元素个数；a_i 是元素 E 的一对子元素标签，如 <element>…</element>, $(1\le i\le n)$。

定义 2 将 $(a_1,a_2,\cdots,a_n)$ 按字典序法[11]生成 $n!$ 个排列，如果排列 $(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 是子元素排列 $(a_1,a_2,\cdots,a_n)$ 的一个新排列，则称 $E(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 是 $E(a_1,a_2,\cdots,a_n)$ 的等价元素。由此可得：

性质 1 元素 $E(a_1,a_2,\cdots,a_n)$ 有 $n!$ 个等价元素(包括 E)。

由于 XML 保持数据存储与数据显示相分离，且可以结合样式语言来定义如何显示 XML 文档中的数据，使用等价元素排列修改后的 XML 文件大小没有改变，数据也没有改变，且不影响 XML 文件在浏览器下的可阅读性，因此

性质 2 等价元素功能相同。

设 $T(E)$ 是 $E(a_1,a_2,\cdots,a_n)$ 等价元素的集合，设集合 $Z(E)=\{z\mid z\in Z,0\le z\le n!-1\}$，$s(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 是 $(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 在 $n!$ 个排列中对应位置的序号，且 $s(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)\in Z(E)$。

定义 3 定义函数 f，$\forall\ E(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)\in T(E)$，$\exists\ p\in Z(E)$，当 $p=s(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 时，函数 $f\left(E(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)\right)=p$ 成立；$\forall\ p\in Z(E)$，$\exists\ s(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)=p$，逆函数 $f^{-1}(p)=E(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$ 成立。

设 XML 文件 X 中，$|X|$ 表示元素个数，$E_j(a_1,a_2,\cdots,a_n)$ 表示 X 的第 j 个元素，$|E_j|$ 表示该元素的子元素个数 $(1\le j\le|X|)$。选取集合 $X(E)=\left\{E_j(a_1,a_2,\cdots,a_n)\middle|\,|E_j|\ge 2,1\le j\le|X|\right\}$，集合 $A(E)=\{|E_j|\,\|\,|E_j|\ge 2,1\le j\le|X|\}$。

4　信息预处理

为增强信息传输的安全性，提高隐藏信息的健壮性，通常应在嵌入前对嵌入对象进行一定预处理。先将待隐藏信息 M 编码成二进制信息序列，然后采用 RSA 算法对二进制序列进行加密。根据信息隐藏的机理，通常可以把信息隐藏系统等效为一个通信系统[12]：原始载体信号可以视为一个宽带信道，从原始载体信号中提取的用于隐藏信息的载体特征视为载波，隐藏信息视为一个调制信号，隐藏信息的嵌入/提取可以模型化为调制/解调，各种攻击可以等效为信道噪声。因此，能采用类似于通信系统中的信道编码思想，通过对隐藏信息进行纠错编码来增强隐藏信息的健壮性。目前常用的差错控制编码主要有前向纠错编码、线性分组码、BCH 码、卷积码、TCM 编码、Turbo 码等。由于在同等编码复杂度下，卷积码具有比分组码更强的纠错能力，因此本文选择卷积编码对加密信息进行编码。最后将编码后的秘密信息二进制序列 M 转换成的十进制整数 N。信息预处理流程见图 2。相应地，在接收端提取出秘密信息 M 后，要进行信息后处理，其流程与预处理相反。

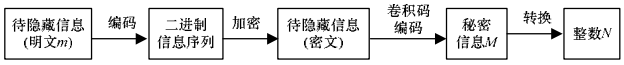

图 2　信息预处理流程

5　相关算法

为算法简洁，定义如下 4 个函数：

(1)预处理函数 *pre_treat*()，对明文进行预处理，得到密文 M；

(2)转换函数 *M_to_N*()，将秘密信息 M 转换为十进制整数 N 的函数；

(3)提取函数 *N_to_M*()，将整数 N 转换为秘密信息 M 的函数；

(4)后处理函数 *late_treat*()，对密文 M 进行后处理，得到明文。

5.1　信息嵌入算法

信息嵌入算法描述如下：

Step1 M= *pre_treat* (m,k)，其中，M, m, k 分别表示密文、明文和密钥；

Step2 N =*M_to_N*(M)；

Step3 令 $i=1, j=1\ (1\le i\le |A(E)|)$；

Step4 如果 $|E_j|\in A(E)$，转 Step5，否则令 $j=j+1$，并返回 Step4；

Step5 令 $N'=N/|E_j|!$, $p_i=N \bmod |E_j|!$，求 $E_j(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)=f^{-1}(p_i)$，并令 $E_j(a_1,a_2,\cdots,a_n)=E_j(\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n)$；

Step6 令 $N=N'$，如果 $N=0$，则嵌入完成，否则令 $i=i+1$, $j=j+1$，并返回 Step4。

5.2 信息提取算法

信息提取算法描述如下

Step1 求 $p_i=f\left(E_j(a_1,a_2,\cdots,a_n)\right)$，并得到集合 $P=\left\{p_1,p_2,\cdots,p_{|A(E)|}\right\}$，其中，$1\le i\le |A(E)|$，$1\le j\le |X|$，$E_j(a_1,a_2,\cdots,a_n)\in X(E)$；

Step2 由迭代公式

$$N_i=\begin{cases}N_{i+1}\times|E_j|!+p_i, 1\le i<|A(E)|, & p_i\in P,|E_j|\in A(E)\\ 0, i\ge|A(E)|\end{cases}$$

计算出 N_1；

Step3 $N=N_1$，N 是嵌入的整数；

Step4 *M*=*N_to_M*(*N*)；

Step5 *m*=*late_treat*(*M*)。

6 信息隐藏容量

某元素的子元素排列组合个数代表可以隐藏信息量的多少，因此，可以将其定义为该元素的最大隐藏量[11]。设 $|E|=n$，M_E 分别表示元素 $E(a_1,a_2,\cdots,a_n)$ 的子元素个数和最大隐藏量。根据等价元素的性质 1，可得

$$M_E=|E|!=n! \tag{1}$$

设 B_E 是元素 $E(a_1,a_2,\cdots,a_n)$ 能够隐藏信息的最大比特数，则

$$B_E=\mathrm{lb}(n!) \tag{2}$$

设 M_X 是 XML 文件 X 的最大隐藏量，由式(1)可知，有 $|E_j|$ 个子元素的元素最大隐藏量是 $|E_j|!$。从嵌入算法可知，整数 N 不断除以集合 $X(E)$ 中元素的最大隐藏量进行递减。因此，XML 文件 X 的最大隐藏量是集合 $X(E)$ 中元素的最大隐藏量的乘积，即

$$M_X=\prod\left(|E_j|!\right).|E_j|\in A(E) \tag{3}$$

设 B_X 是 XML 文件 X 能够隐藏信息的最大比特数，则

$$B_X=\mathrm{lb}\prod\left(|E_j|!\right),|E_j|\in A(E) \tag{4}$$

7 实验与分析

7.1 隐藏容量

比较本算法与算法 1、算法 2、算法 3 的隐藏容量，选用图 1 所示的 XML 文档作为隐藏载体。本算法的信息嵌入量以 2 为底，集合 *X*(*E*)中的元素对应子元素排列组合的积的对数。算法 1 通过改变标记之间尺寸来隐藏信息。算法 2 通过对文件中的元素名称进行同义词替换实现信息隐藏，假设每组同义词的个数为 $n(n\ge 2)$。算法 3 利用引号作为信息隐藏的载体，XML 文档中只有在属性命名时才用到引号，但子元素也可以利用元素属性来表示。为了计算算法 3 的最大隐藏量，先将图 1 所示的 XML 文档的子元素用属性来表示，例如，将 <person><name>JAM</name> 改为 <persion name= “JAM”>，然后计算算法 3 的最大隐藏量。4 种算法的信息隐藏容量比较如表 1 所示。

表 1 4 种算法的信息隐藏容量比较

Person 元素个数	最大隐藏比特数				隐藏前后文档大小是否发生变化			
	本算法	算法 1	算法 2	算法 3	本算法	算法 1	算法 2	算法 3
2	8	8	12×lb*n*	8	否	是	是	否
3	14	12	18×lb*n*	12	否	是	是	否
4	19	16	24×lb*n*	16	否	是	是	否
8	44	32	48×lb*n*	32	否	是	是	否
16	102	64	96×lb*n*	64	否	是	是	否

当 XML 文档中元素个数增加时，本算法的最大隐藏量递增的速度远超过元素个数递增的速度，其他算法不具有该优势。由表 1 可以看出，本算法的最大隐藏量优于算法 1 和算法 3。当 XML 文档中元素个数较少时，本算法的信息隐藏量劣于算法 2，但算法 2 需要先构造同义词库，隐藏容量在很大程度上依赖于每组同义词的个数，且在 XML 文件中某一元素的兄弟元素与其自身常为同一类型元素，为了保证隐蔽性，只能选择其中的部分元素作为隐藏载体，因此，会降低隐藏量。算法 3 利用引号作为信息隐藏的载体，编码时单引号表示 0，双引号表示 1。但 XML 文档中只有在属性命名时才用到引号，且元素属性可以利用子元素来表示，因此，隐藏容量难以保证。

7.2 隐蔽性

分别以本算法、算法 1 和算法 2 向图 1 所示的 XML 文档嵌入秘密信息“101”，观察其隐蔽性。

用本算法隐藏信息后的 XML 文档描述如下：

```
<?xml version="1.0"?>
<people>
  <person>
   <name>JAM</name>
   <telephone>
     <office>99-12</office>
     <home>55-12</home>
     <mobile>123</mobile>
   </telephone>
  </person>
     …
  </people>
```

用算法 1 隐藏信息后的 XML 文档描述如下：

```
<?xml version="1.0"?>
<people>
  <person>
   <name>JAM    </name>        “1”
   <telephone>
     <home>55-12</home>       “0”
     <office>99-12   </office>   “1”
     <mobile>123</mobile>
   </telephone>
  </person>
      …
  </people>
```

用算法 2 隐藏信息后的 XML 文档描述如下：

```
<?xml version="1.0"?>
<people>
  <PErson>                  “1”
   <name>JAM</name>     “0”
   <TElephone>              “1”
```

```
    <home>55-12</home>
    <office>99-12</office>
    <mobile>123</mobile>
  </TElephone>
</PErson>
  …
</people>
```

可以看出，本算法只修改了 XML 文档中子元素的排列顺序，不修改元素标记和文档的内容信息，信息隐藏后文档长度不变。且等价元素与整数间的映射关系可以随机指定，因此，隐蔽性较好。算法 1 在部分元素的起始标记之间增加了不可见字符，因此，增加了文件长度，容易被发现。算法 2 将部分元素名称用同义词进行替换(本实验通过改变字母大小写形成同义词)，通过类似 ULTRAEDIT 的编辑软件打开时，很容易引起用户怀疑。算法 3 利用引号作为信息隐藏的载体，但单引号、双引号的频繁更替可能引起对方注意，降低了隐蔽性。

7.3 鲁棒性

由于 XML 文档所含冗余信息较少，因此基于 XML 的信息隐藏方法的鲁棒性弱于多媒体信息隐藏。在不同类型的攻击下，XML 文档中所隐藏信息的完整性反映了该算法的鲁棒性。比较本算法的鲁棒性与另外 3 种算法的鲁棒性，结果如表 2 所示，其中，“ 完整 ”表示隐藏信息没有丢失或错误 ;“ 不完整 ” 表示隐藏信息有部分丢失或错误。

表 2　4 种算法的鲁棒性

算法	文档格式攻击	删除/篡改元素名称	删除/篡改元素内容	篡改元素排列顺序	删除/篡改元素属性
本算法	完整	完整	完整	不完整	完整
算法 1	不完整	不完整	完整	不完整	完整
算法 2	完整	不完整	完整	不完整	完整
算法 3	完整	完整	完整	不完整	不完整

在本算法中，信息的隐藏只与元素排列有关。由表 2 可以看出，当文档中的元素名称、元素内容和元素属性受到删除或篡改攻击，或文档受到一般格式攻击后，本算法隐藏的信息不会丢失。当文档中的元素顺序受到篡改攻击后，隐藏的信息将会发生差错或部分丢失，解决方法可以采用信息预处理中的差错控制编码。

8　结束语

信息隐藏技术已成为网络时代信息安全领域的新热点，它被广泛用于国家安全、电子商务、电子政务、版权保护、隐蔽通信等领域[11,13]。本文提出的信息隐藏算法对待隐藏信息进行预处理，使其转换为一个大的整数，并利用字典法形成等价元素的集合，进而通过等价元素相互置换来隐藏信息。

该算法具有如下应用意义：(1)将网站的说明信息或版权信息藏于页面文档中，以防止一般性的恶意复制或转载，起到保护网页的作用；(2)将隐秘信息藏于网页中，特定用户可以访问该网页并提取隐秘信息，从而实现隐蔽通信。

参考文献

[1] Petitcolas F A P, Anderson R J, Kuhn M G. Information Hiding——A Survey[J]. Proceedings of the IEEE, 1999, 87(7): 1062-1078.

[2] Johnson F, Jajodia S. Steganography: Seeing the Unseen[J]. IEEE Computer, 1998, 31(2): 26-34.

[3] Hartung F, Kutter M. Multimedia Watermarking Techniques[J]. Proceedings of the IEEE, 1999, 87(7): 1079-1107.

[4] Moulin P, O'Sullivan J A. Information-theoretic Analysis of Information Hiding[J]. IEEE Transactions on Information Theory, 2003, 49(3): 563-593.

[5] 华宇宁, 杨　颖, 李青川. 基于 LabVIEW 的图像水印系统[J]. 南京理工大学学报, 2008, 32(3): 300-303.

[6] 刘光杰, 戴跃伟, 赵玉鑫, 等. 隐写对抗的博弈论建模[J]. 南京理工大学学报, 2008, 32(2): 199-204.

[7] Provos N, Honeyman P. Hide and Seek: An Introduction to Steganography[J]. IEEE Security & Privacy, 2003, 1(3): 32-44.

[8] 吴　晶, 王书文. 基于 XML 语言的信息隐藏方法[J]. 中国安全科学学报, 2005, 15(12): 78-80.

[9] 丁光华, 刘嘉勇, 孙克强. 基于 XML 的信息隐藏方法[J]. 计算机工程, 2008, 34(6): 155-157.

[10] 周　莉, 王炼红, 李丽娟. 一种基于 XML 文档的数字水印方案[J]. 湖南大学学报, 2007, 34(5): 83-86.

[11] 孙星明, 黄华军, 王保卫, 等. 一种基于等价标记的网页信息隐藏算法[J]. 计算机研究与发展, 2007, 44(5): 756-760.

[12] 王育民, 张　彤, 黄继武. 信息隐藏——理论与技术[M]. 北京: 清华大学出版社, 2006.

[13] 眭新光, 罗　慧. 一种安全的基于文本的信息隐藏技术[J]. 计算机工程, 2004, 30(19): 104-105.

编辑　陈　晖

(上接第 152 页)

密钥，U 相信 S 相信 K_{us} 是适合适合双方通信的确认共享会话密钥。

4　结束语

本文提出一种适于受限环境的认证协议，可以减少客户端的计算量、提高执行效率。该协议实现了预期安全目标，并通过了扩展 SVO 逻辑的证明。下一步工作将在此基础上总结扩展 SVO 逻辑的原则问题，从而在一定程度上避免需要增加的初始假设。

参考文献

[1] Syverson P F, Vanoorscho P C. On Unifying Some Cryptographic Protocol Logics[C]//Proceedings of the 1994 IEEE Computer Society Symposium on Security and Privacy. Los Alamitos, USA: IEEE Computer Society Press, 1994: 14-28.

[2] Syverson P F, Vanoorscho P C. A Unified Cryptographic Protocol Logics[R]. Washington D. C., USA: Naval Research Laboratory, Technical Report: 5540-227, 1996.

[3] Burrows M, Abadi M, Needham R. A Logic of Authentication[J]. ACM Transactions on Computer Systems, 1990, 8(1): 18-36.

[4] 莫　燕, 张玉清, 李学干. 关于安全协议设计原则的研究[J]. 计算机工程, 2005, 31(24): 183-185.

[5] 卿斯汉. 安全协议[M]. 北京: 清华大学出版社, 2005.

编辑　陈　晖