武汉大学学报(信息科学版)
Geomatics and Information Science of Wuhan University
ISSN 1671-8860,CN 42-1676/TN

《武汉大学学报(信息科学版)》网络首发论文

题目：面向 OpenDRIVE 格式高精地图的脆弱水印法
作者：袁天洋，朱长青，陈会仙，任娜，吕旭超
DOI：10.13203/j.whugis20230500
收稿日期：2024-04-18
网络首发日期：2024-05-15
引用格式：袁天洋，朱长青，陈会仙，任娜，吕旭超．面向 OpenDRIVE 格式高精地图的脆弱水印法[J/OL]．武汉大学学报(信息科学版).
https://doi.org/10.13203/j.whugis20230500

DOI: 10.13203/j.whugis20230500
引用格式：

袁天洋, 朱长青, 陈会仙,等. 面向 OpenDRIVE 格式高精地图的脆弱水印法[J].武汉大学学报（信息科学版），2024，DOI: 10.13203/j.whugis20230500（YUAN Tianyang, ZHU Changqing, CHEN Huixian, et al. Fragile Watermarking Algorithm for High Precision Map of OpenDRIVE Format[J].Geomatics and Information Science of Wuhan University, 2024 , DOI: 10.13203/j.whugis20230500）

面向 OpenDRIVE 格式高精地图的脆弱水印法

袁天洋[1,2,3] 朱长青[1,2,3*] 陈会仙[4] 任娜[1,2,3,5] 吕旭超[1,2,3]

1 虚拟地理环境教育部重点实验室(南京师范大学)，江苏 南京，210023

2 江苏省地理环境演化国家重点实验室培育建设点，江苏 南京，210023

3 江苏省地理信息资源开发与利用协同创新中心，江苏 南京，210023

4 自然资源部地图技术审查中心，北京，100830

5 湖南省地理信息安全与应用工程研究中心，湖南 长沙，410017

摘　要：针对高精地图数据完整性保护问题，本研究基于零宽度字符和信息-摘要算法提出了一种适用于 OpenDRIVE 格式高精地图的脆弱水印算法。嵌入水印过程中，先按照文档结点的树状结构与道路要素的起始点坐标排序建立位置关系，再提取结点外部文本生成脆弱水印信息，最后结合建立的位置关系将水印信息嵌入到对应结点。在验证时，对比生成的水印信息与提取的水印信息，根据两者差别鉴定数据完整性。实验表明，该算法生成的水印不可见性良好，不会改变地图所包含的信息，造成的增量与原始大小的比值通常可控制在 5%以内，验证水印时能够精准定位文档中发生的篡改。

关键词：脆弱水印；高精地图；零宽度字符；XML；文本水印

Fragile Watermarking Algorithm for High Precision Map of OpenDRIVE Format

YUAN Tianyang[1,2,3] *ZHU Changqing*[1,2,3*] *CHEN Huixian*[4] *REN Na*[1,2,3,5] *LYU Xuchao*[1,2,3]

1 Key Laboratory of Virtual Geographic Environment Ministry of Education, Nanjing Normal University, Nanjing 210023, China

2 State Key Laboratory Cultivation Base of Geographical Environment Evolution, Jiangsu Province, Nanjing 210023, China

3 Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

4 Map Supervision Centre, Ministry of Natural Resources, Beijing 100830, China

收稿日期：2024-04-18

项目资助：国家自然科学基金（42071362）。

第一作者：袁天洋，硕士生，主要研究方向为测绘地理信息安全。221302129@njnu.edu.cn

通讯作者：朱长青，教授。649397417@qq.com

5 Hunan Engineering Research Center of Geographic Information Security and Application, Changsha 410017, China

Abstract: Objectives: It is not supposed to be ignored that the issue of data integrity protection of high precision map, which is considered as a key data resource in the development of digital transportation. However, hash algorithms applied to file verification did not consider the function of inferring the location of tampered content inside a message based on a single checksum exception in its design. Therefore, an algorithm that can verify the integrity of high-precision map data and accurately locate tampering can not only effectively prevent and resolve network security risks in the field of digital transportation, but also effectively protect the safety of people's lives and property. Focusing on this question, this paper proposed a fragile watermarking algorithm that is suitable for high precision map of OpenDRIVE format. **Methods:** This algorithm is designed based on Unicode zero-width characters and MD5 hash algorithm. The algorithm proposed use zero-width character sequences to embed watermark information, with XML nodes as units to generate, embed, extract and verify watermarks. The watermark information corresponding to each node is composed of a combination of a tree watermark and a road feature watermark. In order not to affect the normal use of the data, the embedding position of the watermark sequence is selected at the end of the row corresponding to the starting label or unique label of each node. When the data is distributed, the administrator can use this algorithm to generate fragile watermarks and embed them into high-precision map data and verify whether the watermarks are successfully embedded in the file. When the user needs to verify the integrity of the file, this algorithm can be used to extract and verify the watermark. If the two are completely consistent, inform the user that the integrity of the file has passed the inspection, otherwise the algorithm will determine the tampering location for the user based on the abnormal situation. **Results:** The result show that: (1) Embedding watermarks into high-precision map documents will not cause visible abnormal display phenomena. (2) Embedding watermarks will not cause significant changes in the size of high-precision map documents, and the ratio of increment to original size can be controlled within 5%. (3) Highly sensitive to attacks against nodes and watermarks, and able to accurately locate the location of tampering. When whole element deletion occurs, the geometric information of the deleted element can be inferred. **Conclusions:** This algorithm is suitable for the integrity protection of high-precision map data, and can also be applied to the integrity protection of other structurally similar data such as HTML and CSS, as long as cancelling some preprocessing requirements, reconfirming the appropriate watermark embedding position and clarifying the definition of features and the sorting rules between features.

Key Words: Fragile Watermarking；High Precision Map；Zero-Width Characters；XML；Text Watermarking

“十三五”期间，我国交通运输行业信息化数字化取得了长足发展[1]。高精地图是一种主要用于高级别辅助驾驶和智能驾驶的专用电子地图[2]，是数字交通发展中的关键数据资源。其为对普通导航地图的颠覆性升级，服务主体为机器而非人类，部分应用情境下制作与使用同步实时，不只是服务于导航，还包括导航决策本身，是驾驶环境的数字孪生体，具有高精度、高丰富度、高动态性等特点，在自动驾驶中扮演着不可或缺的指挥员角色[3]。在数据分发、共享的过程中，高精地图数据可能被无意或恶意篡改。然而，应用于文件校验的杂凑算法在设计时未考虑实现根据单一校验码异常的情况推断消息内部被篡改内容所在位置的功能[4]。因此，一种可以验证高精地图数据完整性，精准定位篡改的技术手段，能够有效防范化解数字交通领域的网络安全风险，切实保护人民生命财产安全。

数字水印技术是一种能将版权和用户

信息作为水印内容隐蔽地嵌入到数据中，并与数据融为一体，成为数据不可分离的一部分的技术[5]。其中，鲁棒水印主要用于版权保护与溯源追踪，而脆弱水印主要用于数据的真伪辨别和完整性鉴定。脆弱水印能够有效认证出数据是否被篡改、何处被篡改、篡改到何种程度，从而保证数据可靠、可信、可用[6]。尽管检测鲁棒水印过程中产生的中间参数也可用于推断数据完整性是否被破坏，但是这一方法无法定位篡改发生的位置，且存在漏检篡改的风险。故面对完整性验证问题，脆弱水印方案更具优势。

虽然专门适用于高精地图数据的数字水印算法少有研究，但是 OpenDRIVE 格式高精地图数据作为一种符合可扩展标记语言（extensible markup language，XML）规范[7]的非格式化文本数据[8]，仍有不少对应的文本水印算法可供参考。这些算法可归为如下四类：

（1）基于自然语言处理的水印算法。这类算法通过对文本中的短语、词汇、字母进行等价或近似的替换，实现水印的嵌入。王炳锡等人[9]、Li[10]和 Mir[11]等人提出的水印算法适用于面向人类的文本应用场景，但在面向机器的文本应用场景中，会严重影响文档可用性。蔡毅等人[12]、姚荣华等人[13]和陈丽等人[14]提出的算法充分利用了 XML 文档标签属性不依赖大小写的特性，不会增加存储、传输文档的负担，但水印容量严重依赖标签英文字符数量，不可见性较差，且算法不抗大小写转换攻击。

（2）基于词频统计的零水印算法。斯琴等人[15]和 Al-Wesabi[16]等人设计的文本水印算法不会改动原始文本，能够抵抗常见攻击，但是需要搭建或租用版权保护服务器，对使用者网络环境有一定的要求。

（3）基于零宽度字符的水印算法。李兆璨等人[17]、陈旖旎等[18]和彭登[19]将水印信息转化为零宽度字符序列并嵌入在文本中。这类算法生成的水印不可见性良好，水印容量不受原始文本限制，不会影响机器读取数据。但是若不对水印序列长度进行限制，过长的水印序列极易使得文档大小剧烈变化，严重占用存储资源，影响读取和传输效率。此外，有的零宽度字符在使用特定软件浏览时确实具有不可见性，但是更换软件浏览则可能会有肉眼可察觉的异常显示现象。

（4）基于字符变形和字库替换的水印算法。Xiao C 等人[20]、Qi W 等人[21]设计变形拉丁字符并向英文字符编码库中添加编码，孙杉等人[22]、姚晔等人[23]设计变形汉字字符并向中文字符编码库中添加编码，再利用变形字符替换正常字符的方式嵌入水印。这种方法不可见性良好，不会增加存储开销。不过在面向机器读取的情境下，机器需要额外安装含变形字符的编码库以正确读取数据，这会让用户增加额外的软件开销。

综上所述，现有相关算法有各自的优势和不足，且多为鲁棒水印算法，少有脆弱水印算法。更为关键的是，现有相关算法应用于高精地图文档时，不能同时满足以下三个需求：

（1）向高精地图文档中嵌入水印后，无肉眼可见的异常显示现象。

（2）嵌入水印造成的文件大小变化有限且可控。

（3）精准定位篡改位置，发生整要素删除时，推断被删除要素的几何信息。

为了满足上述需求，本研究以零宽度字符为水印序列元素，以信息-摘要算法（message-digest algorithm 5，MD5）为数学基础，针对高精地图数据特性提出了一种脆弱水印算法。

1　面向高精地图的脆弱水印算法

1.1 总体思想

面向 OpenDRIVE 格式高精地图的脆弱水印算法（以下简称“高精地图脆弱水印算法”）以零宽度字符序列为基础，XML 结点为单位生成、嵌入、提取和验证水印信息。每一结点对应的水印信息都由树状水印和

道路要素水印组合而成。为了不影响数据的正常使用，水印序列的嵌入位置选定在每一个结点对应的起始标签或唯一标签的所在行行末。经检验，此处嵌入水印信息不会影响自动驾驶设备读取高精地图数据。

当数据被分发时，发行方可利用本算法生成脆弱水印并将水印嵌入到高精地图数据中并验证水印是否成功嵌入。需要验证文件完整性时，使用方可利用本算法提取与并验证水印。如果两者完全一致，则告知文件完整性通过检验，否则算法根据异常情况确定篡改位置。

1.2 水印序列元素

为了改进传统的基于零宽度字符的文本水印在更换软件浏览后易出现异常显示的问题，本研究设计了不可见性测试，以寻找不可见性更好的零宽度字符。通过测试的零宽度字符将作为组成高精地图脆弱水印序列的零宽度字符元素。表 1 是查阅统一码（Unicode）官方文档后确定的部分零宽度字符的编号以及含义。以上所有零宽度字符分别组成只包含单一字符且长度不小于 20 的序列，再各自嵌入一段文本，并依次在 Windows 记事本、Visual Studio、Chrome 内核的浏览器、Notepad++等多种常用于查看和编辑非格式化文本的软件中打开，观察嵌入前后其显示效果是否出现可见的变化。只有三种零宽度字符在所有测试到的常用文本编辑器与浏览器中均不会引发任何显示上的异常。将它们选作水印序列元素，为后续表述方便，依次对应为三进制数 0、1 和 2，如表 2 所示。

表 1 部分零宽度字符及含义

Tab.1 Some of Zero-Width Characters With Meanings

Unicode 编码	含义	Unicode 编码	含义
U+180E	蒙文元音分隔符	U+202B	嵌入式从右至左标记
U+200B	零宽度空格	U+202C	文字方向变化结束
U+200D	零宽度连接符	U+202D	强制从左至右标记
U+200E	成对从左至右标记	U+202E	强制从右至左标记
U+200F	成对从右至左标记	U+206A	禁止对称交换
U+202A	嵌入式从左至右标记	U+FEFF	零宽度无间断空格

表 2 通过多平台不可见性测试的零宽度字符

Tab.2 Zero-Width Characters Passed in Multi-Platform Invisibility Testing

Unicode 编码	原含义	对应三进制数
U+200B	零宽度空格	0
U+180E	蒙文元音分隔符	1
U+FEFF	零宽度无间断空格	2

1.3 水印生成

由于组成水印序列的零宽度字符仅有 1.2 节所确定的三种，每个零宽度字符所能承载的信息量较小。为了在这一前提下生成不仅具有序列随机性、解码结果唯一性，而且嵌入后不造成文件大小剧烈变化的水印序列，本算法在水印生成环节采取两种措施解决问题：

（1）截取哈希值低位，根据所保护的内容长度及重要性缩短水印信息长度，以此缩短水印序列长度。

（2）使用三进制最优前缀编码映射十六进制哈希值，保证解码结果唯一性前提下尽可能缩短水印序列。各十六进制数对应的编码如表 3 所示。这些编码的本质是当前频率下最优三叉树的叶子结点，如图 1 所示。

表 3 十六进制数字最优前缀编码表

Tab.3 Prefix Code of Hexadecimal Digits

数字	编码	数字	编码
0	00	8	101

1	010	9	102
2	011	A	110
3	012	B	111
4	020	C	12

5	021	D	20
6	022	E	21
7	100	F	22

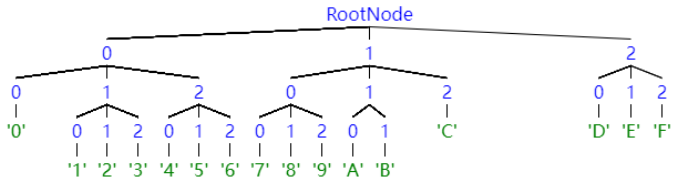


图 1 十六进制数字最优三叉树示意图

Fig. 1 Optimal Ternary Tree of Hexadecimal Digits

为了更好地检测和定位篡改，本算法嵌入的每一处水印信息都由两部分组成。

第一部分，树状水印。其长度关于深度递减，关于结点文本长度递增，直至为 0。树状水印能有效避免算法在单处篡改且篡改导致节点局部发生哈希碰撞的极端情况下完全漏检篡改。以下是具体生成方式。

根据结点 r 外部文本 O_r 生成对应的哈希值 $\{v_r\}$。$\{v_r\}$ 中低位 h 字节信息记为 $\{v_{r,h}\}$。编码 $\{v_{r,h}\}$，得到零宽度字符序列 $\{m_{r,h}\}$。对于不同的结点，h 取值也不同。对于序号为 r 的结点，该节点树状水印信息字节数 h_r 取值与该结点深度 d_r 以及该结点文本长度 l_r 关系为：

$$h_r=\begin{cases}8, d_r=0\\ 1, 0<d_r\le 2\wedge l_r\ge 4000\\ 0.5, d_r=1\wedge l_r<4000\\ 0.5, d_r=2\wedge 500\le l_r<4000\\ 0, d_r>2\vee d_r=2\wedge l_r<500\end{cases} \tag{1}$$

若 r 为道路要素结点，根据文件头结点外部文本 O_{hdr} 生成的哈希值 v_{hdr} 第 53 至第 56 位生成文档标记 m_{hdr}，并连接在 $m_{r,h}$ 后。若 r 为除道路要素结点以外的其它要素结点，且为同标签名结点中首个被遍历到的结点。设标签名为 Ln 的所有要素结点的外部文本依遍历序依次连接得到的文本为 O'_{Ln}，据 O'_{Ln} 生成的哈希值为 v_{Ln}，由 v_{Ln} 低位 1 字节生成的非道路要素标记为 m_{Ln}，将 m_{Ln} 连接在 $m_{r,h}$ 后。如此，便得到树状水印 m_r。

第二部分，道路要素水印，对于描述道路要素的结点，道路要素水印信息字节数 $h'=2$，对于其余结点，其为空白串，即 $h'=0$。设计道路要素水印的目的是允许算法帮助用户在道路要素被整要素删除的情况下，也能推断出被删除的道路要素大致位置。具体生成方式如下：

获取道路要素集合 $\{R_i\}$ 在参考线起点处惯性坐标系的坐标值 $\{(x_i,y_i)\}$，分别对横坐标 $\{x_i\}$、纵坐标 $\{y_i\}$ 由小到大排序，排序结果为 $\{x_i^{'}\}$ 与 $\{y_i^{'}\}$。根据这一结果确定对于每一个道路要素对应的结点，哪些文本用于生成该结点对应的道路要素水印。道路要素水印由四个水印信息字节数为 0.5 的部分组成：由横坐标排序(x)中的前一结点(f)确定的 $m'_{r,x,f}$、后一结点(l)确定的 $m'_{r,x,l}$、由纵坐标排序(y)中的前一结点确定的 $m'_{r,y,f}$ 以及后一结点确定的 $m'_{r,y,l}$。对于每个道路要素 R_i，记其对应的结点为 r_i，则 R_i 对应生成 $m'_{r,x,f}$ 的输入文本 $T_{i,x,f}$ 和 $T_{i,x,l}$，有：

$$T_{i,x,f}=\begin{cases}O_{r_i}, I(x_i^{'})=1\\ O_{r(I(x_i^{'})-1)}, I(x_i^{'})\ne 1\end{cases} \tag{2}$$

$$T_{i,x,l}=\begin{cases}O_{r_i}, I(x_i^{'})=c\\ O_{r(I(x_i^{'})+1)}, I(x_i^{'})\ne c\end{cases} \tag{3}$$

其中，$I(x_i^{'})$ 为 R_i 在横坐标排序结果中

的位置，$r(I(x_i^{'})+1)$为起始点横坐标序号为$I(x_i^{'})+1$的道路要素对应的结点，c为道路要素总数。

$T_{i,x,f}$、$T_{i,x,l}$生成的哈希值分别为$v'_{r,x,f}$、$v'_{r,x,l}$，编码其低位1字节的高位0.5字节，即得到$m'_{r,x,f}$与$m'_{r,x,l}$。依照同样的规则，可得$T_{i,y,f}$和$T_{i,y,l}$，编码其哈希值$v'_{r,y,f}$与$v'_{r,y,l}$低位0.5字节，即得到$m'_{r,y,f}$与$m'_{r,y,l}$，依次连接$m'_{r,x,f}$、$m'_{r,x,l}$、$m'_{r,y,f}$与$m'_{r,y,l}$，得到道路要素水印m'_r。

将树状水印$\{m_r\}$与道路要素水印$\{m'_r\}$依次连接，即生成最终嵌入到文件中的水印序列$\{m''_r\}$。

1.4 水印嵌入

高精地图脆弱水印算法嵌入水印的操作可分为四个步骤，流程如图2所示。

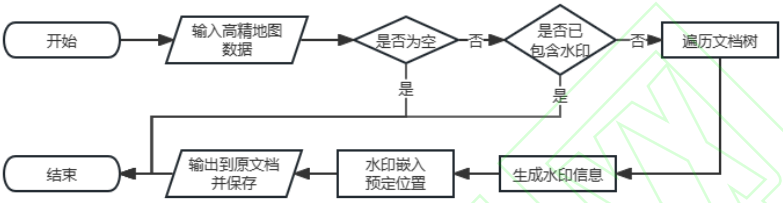


图2 水印嵌入流程图

Fig. 2 Watermark Embedding Flowchart

（1）合法性判断和数据预处理。如果不能读取到文档根节点，则判定数据非高精地图文档，输出提示，并拒绝处理文档。在数据合法的前提下，为水印内容稳定，需要判断是否已有水印存在，如存在，输出提示并拒绝处理文档。

（2）获取生成水印所需信息。获取文档根结点并遍历整个文档树，获取生成水印所需全部结点的信息$\{F_r \mid r=1,2,...,c, F_r=\{N_r, n_r, d_r, O_r\}\}$，其中，$c$为文档树中结点总数，$r$为深度优先遍历文档树时，遍历到某一结点的序号，$N_r$、$n_r$、$d_r$、$O_r$分别为序号为$r$的结点的起始行号，标签名、深度、外部文本。

（3）利用$\{F_r\}$生成水印序列集合$\{m_r\}$。

（4）嵌入水印并保存。将水印序列集合$\{m_r\}$中全部水印序列嵌入到预定位置，即序号为r的结点对应的起始标签或唯一标签所在行行末。最后将更改保存至原地图文档中。

1.5 水印验证

高精地图脆弱水印算法验证水印的操作可分为以下4个步骤，流程如图3所示。

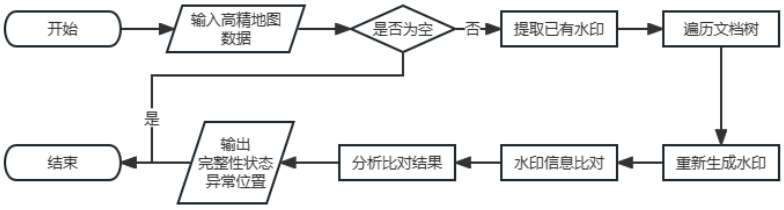


图3 水印验证流程图

Fig. 3 Watermark Validating Flowchart

（1）预验证操作。如果不能读取到文档根节点，则判定数据非高精地图文档，输出提示，并拒绝处理文档。在数据合法的前提下，验证开始前，先提取已有水印并生成除去水印的副本。提取到的已有水印信息记为$\{M_s \mid s=1,2,...,L, M_s=\{N_s, m_s\}\}$，其中$s$为提取水印信息的序号，$L$为提取水印信息总数，$N_s$、$m_s$分别为提取水印序列所在行行号、水印序列内容。

（2）根据文档非水印内容重新生成参考水印信息。打开无水印副本，获取文档根结点并遍历整个文档树，获取生成水印所需全部结点的信息$\{F_r\}$，根据$\{F_r\}$生成水印序列集合$\{m_r\}$，过程同 2.2 节所述，最后删除无水印副本。将$\{m_r\}$中所有水印与其嵌入位置关联，得到参考水印信息$\{M'_s\}=\{N'_s, m'_s\}$，其中s为参考水印信息的序号，L'为参考水印信息总数，N'_s、m'_s分别为参考水印序列应嵌入位置行号、参考水印序列内容。

（3）水印信息比对。逐条比对提取水印信息$\{M_s\}$与参考水印信息$\{M'_s\}$。如果某一行对应水印信息不一致，即证明发现文件异常。输出不一致的水印对应的 XML 结点标签名、起始行行号与异常类型。并输出异常所处的要素类型与 ID 号。

（4）分析比对结果。在本结点起始点位置排列中的前一结点与后一结点的树状水印与要素水印均存在的前提下：

①如果文件被增加一个结点，取决于该结点是从其它含水印文档中截取而来还是篡改者自行编造出来，该结点树状水印文档标记部分异常或树状水印缺失，其所有亲代结点水印树状水印异常。若该结点为要素结点，其要素水印异常，且在起始点坐标排序中，前一结点要素水印中后一结点相关部分异常；后一结点要素水印中前一结点相关部分异常。若为要素结点的后代结点，所在要素结点的树状水印异常，且其所在要素结点在起始点坐标排序中，前一结点要素水印中后一结点相关部分异常；后一结点要素水印中前一结点相关部分异常。

②如果文件被删除一个结点，如果该结点为道路要素结点，则在起始点坐标排序中，原前一结点要素水印中后一结点相关部分异常；原后一结点要素水印中前一结点相关部分异常。若为道路要素结点的后代结点，则其所在要素结点的树状水印异常。且其所在要素结点在起始点坐标排序中，其前一节点要素水印中后一结点相关部分异常；后一结点要素水印中前一结点相关部分异常。如果该结点为非道路要素结点，取决于被删除结点是否为第一个被遍历到的同标签名要素结点，该类型非道路要素标记缺失或内容异常。如果该结点为非道路要素结点的后代结点，则其所属的要素结点树状水印异常。

③如果文件有一个含水印的结点内容发生变动，那么该结点及所有含水印的亲代结点的树状水印异常。如该结点在道路要素结点中，则其所在要素结点在起始点坐标排序中，其前一节点要素水印中后一结点相关部分异常；后一结点要素水印中前一结点相关部分异常。

若该结点水印信息总长为 0，或不存在前一结点或后一结点，则无相应异常信息。

（5）输出结果。若无异常，输出完整性通过验证的信息。若有异常，输出发生异常的要素信息，如有需要，可将具体的水印异常情况一并输出以辅助研判。如遭遇整要素删除，则输出推断的被删除要素大致范围。

2 实验及分析

为了验证由高精地图脆弱水印算法生成的高精地图脆弱水印是否达到预定设计要求，从不可见性、对篡改敏感性以及含水印文件大小变化三个角度进行分析。实验数据为描述某交通环岛的高精地图文档 RoundAbout-3Arms.xodr，其可视化效果如图 4 所示。

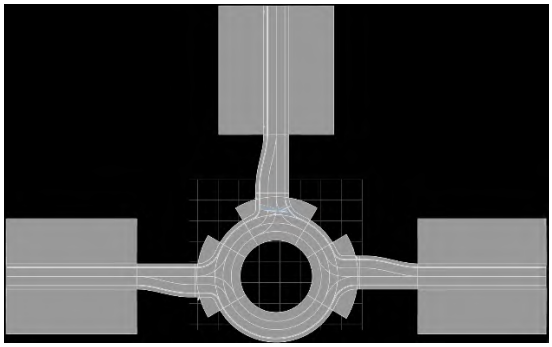


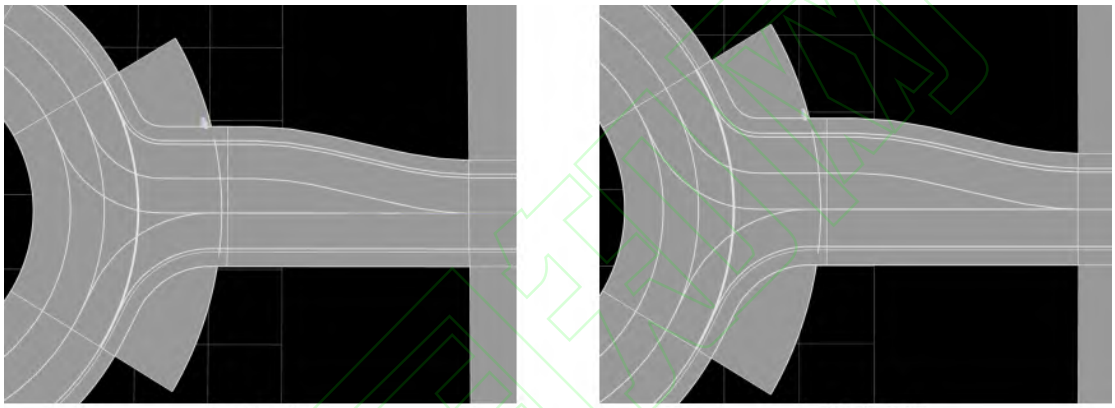
图 4 原始高精地图数据概览图

Fig. 4 Overview Image of Original High Precision Map Data

2.1 不可见性分析

不可见性是指在水印嵌入文本后，数据在视觉层面上无肉眼可见变化的性质。

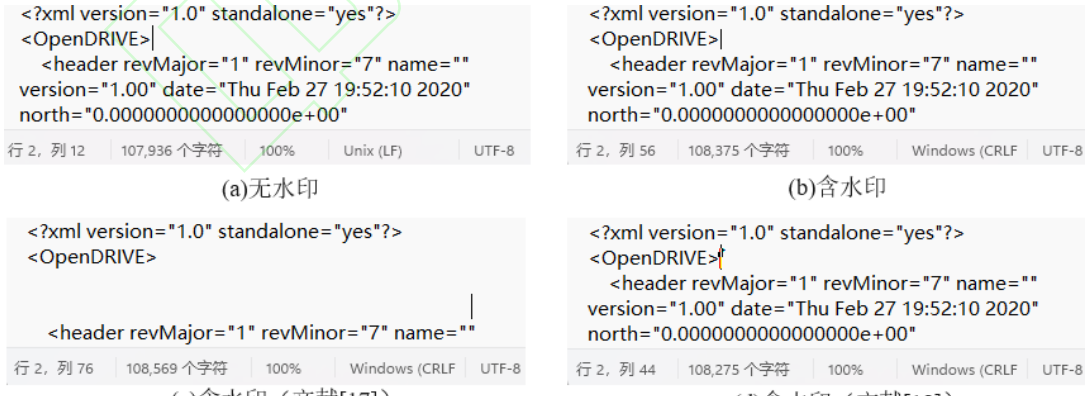
首先对水印嵌入前后的数据可视化效果进行比较，如图 5 所示。图 5(a)是原始数据局部放大的效果，图 5(b)是嵌入水印后与图 5(a)对应区域的可视化效果。由图可见，水印未改变高精地图数据的可视化效果。且本算法不修改属性值，根据属性值所包含的几何信息与属性信息生成的可视化效果理论上与无水印文档一致。因此，在数据可视化效果方面，水印的不可见性良好。



(a)无水印　　(b) 含水印

图 5 局部可视化效果对比

Fig. 5 Comparison of Partial Visualization Effect



(a)无水印　　(b)含水印

(c)含水印（文献[17]）　　(d)含水印（文献[18]）

图 6 Windows 记事本内文本显示效果对比（局部）

Fig. 6 Comparison of Text Display Effect in Software “Windows Notepad” (Partial)

再对文本显示效果进行对比。篇幅所限，以对控制字符最敏感的 Windows 记事本为例，使用本算法嵌入水印后效果如图 6(b)所示，使用文献[17]、[18]所述水印序列元素嵌

入内容相同的水印后效果如图 6(c)、6(d)所示。黑色实心竖线为行末光标位置。由图可见，高精地图脆弱水印虽然改变了行长度，但未改变行末位置，亦未产生额外的显示效果。因此，在文本显示效果方面，水印的不可见性良好。

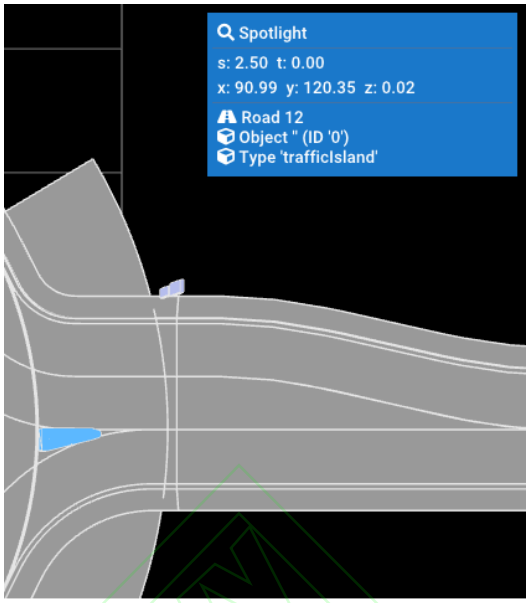
最后，从算法是否对高精地图所承载的地理信息造成改动的角度分析，因为高精地图脆弱水印算法只在特定标签末尾添加零宽度字符序列，不会改变高精地图文档中的任何属性值，所以含水印高精地图文档中包含的所有几何信息以及属性信息都与原始高精地图文档完全一致。因此，在是否改变所载信息角度方面，水印的不可见性亦良好。

2.2 对篡改敏感性分析

攻击者可能在高精地图文档在从分发方将数据传输到用户终端的过程中对含水印的高精地图数据进行篡改。错误路网数据的高精地图可能扰乱自动驾驶车辆对交通环境的感知，造成危害性后果。本节将模拟攻击者可能的攻击手段，验证高精地图脆弱水印算法是否如既定设想检出并定位篡改。

2.2.1 针对道路要素内部结构的攻击

（1）道路要素内部增加结点。向 id 为 12 的道路要素内部增加一个包含障碍物信息的子结点，该子结点为原地图文档中存在的交通岛障碍物结点。篡改后可视化效果与脆弱水印验证结果如图 7 所示。对比验证结果和实际发生的篡改可知，本算法准确定位了被篡改的要素。



(a)篡改后可视化效果

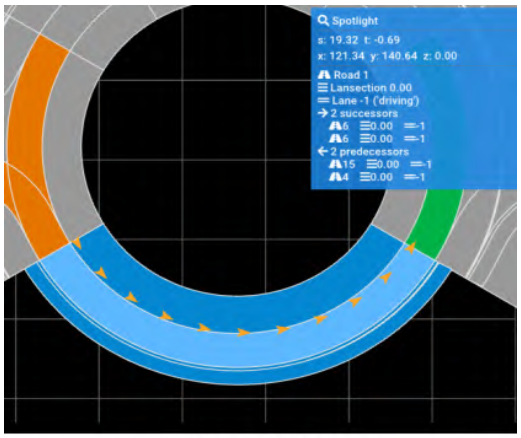
```
异常要素信息：
异常要素起始行：979，要素类型：road，要素ID：12
```

(b)检测结果

图 7 道路要素内部增加结点攻击

Fig. 7 Node Addition Attack Within Road Feature

（2）道路要素内部删除结点。将 id 为 1 的道路要素中 id 为-2 的车道删除。篡改后可视化效果与脆弱水印验证结果如图 8 所示。对比分析结果和实际发生的篡改可知，本算法准确定位了被篡改的要素。



(a)篡改后可视化效果

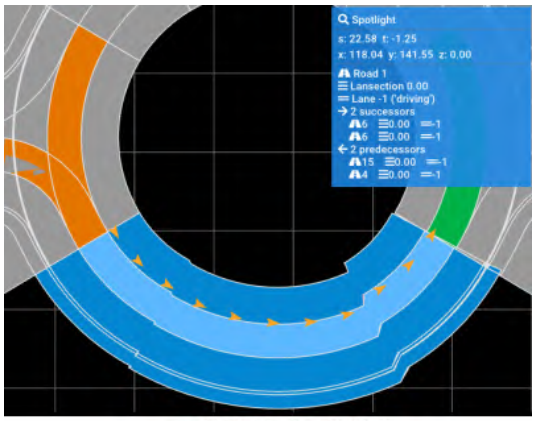
```
异常要素信息：
异常要素起始行：5，要素类型：road，要素ID：1
```

(b)检测结果

图 8 删除结点攻击

Fig. 8 Node Deletion Attack

（3）改动结点内容。篡改 id 为 1 的道路要素中心线参数。篡改后可视化效果与脆弱水印验证结果如图 9 所示。对比验证结果和实际发生的篡改可知，本算法准确定位了被篡改的要素。



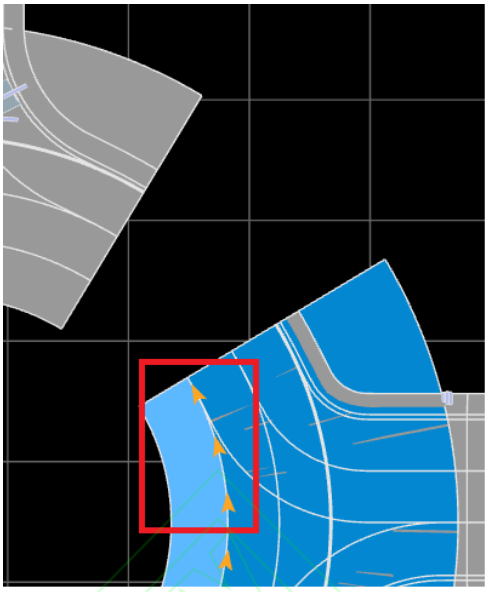
(a)篡改后可视化效果

异常要素信息：
异常要素起始行：5,要素类型：road,要素ID：1

(b)检测结果

图 9 改动结点内容攻击

Fig. 9 Node Contents Altering Attack



(a)篡改后可视化效果和推断篡改位置

异常要素信息：
推断异常要素大致位置在以下四点连线范围内：
(97.7249,109.7036)-(97.7249,120.0000)
-(104.8540,120.0000)-(104.8540,109.7036)

(b)检测结果

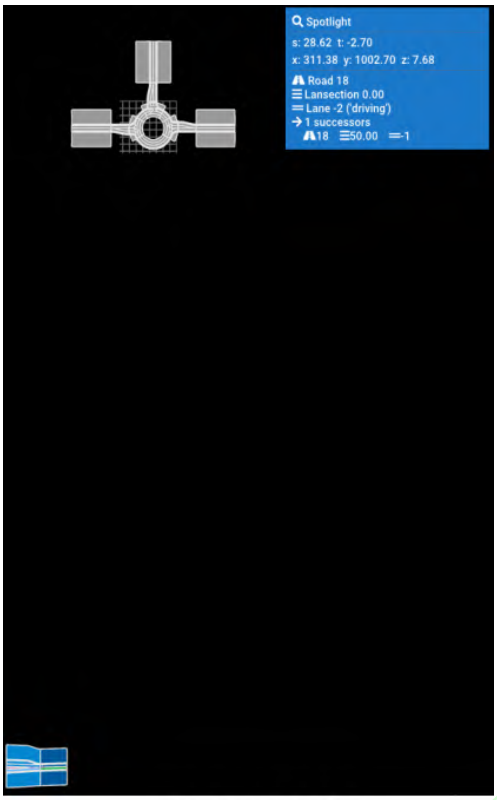
图 10 整要素删除攻击

Fig. 10 Feature Deletion Attack

2.2.2 针对道路要素的攻击

（1）删除道路要素。将 id 为 3 的道路要素删除，如图 10 所示。篡改后原位可视化效果如图 10(a)所示。推断结果如图 10(b)所示，推断被删除要素位置在地图上的位置如图 10(a)中红色矩形框所示。对比验证结果与实际发生的篡改可知，本算法检测到整要素删除，且基本准确地推断了被删除要素的位置。

（2）添加道路要素，将描述某立体交叉路口的含水印高精地图文档 map4.xodr 中 id 为 18 的道路要素添加到 RoundAbout-3Arms.xodr 文档，如图 11 所示。篡改后地图全览效果如图 11(a)所示，蓝色高亮位置为添加的道路要素，推断结果如图 11(b)所示。对比验证结果与实际发生的篡改可知，本算法准确定位到了不属于原文档的要素。



(a)篡改后可视化效果

```
异常要素信息：
异常要素起始行：1436,要素类型：road,要素ID：18
推断异常要素大致位置在以下四点连线范围内：
(156.0132,1000.0000)-(156.0132,1458.4657)
-(468.4638,1458.4657)-(468.4638,1000.0000)
```

(b)检测结果

图 11 添加要素攻击

Fig. 11 Feature Appending Attack

2.2.3 针对水印序列的攻击

针对水印序列的攻击如图 12 所示。

（1）删去水印序列部分内容。将 id 为 10 的道路要素结点对应的水印序列删去三个零宽度字符。验证结果如图 12(a)所示。

（2）移除水印序列。实验中，移除 id 为42的交叉路口要素结点对应的水印序列。验证结果如图 12(b)所示。

（3）交换水印序列位置。实验中，将 id 为 42 的交叉路口要素和 id 为 43 的交叉路口要素的水印序列交换位置。脆弱水印验证结果如图 12(c)所示。

对比分析结果和实际发生的篡改可知，本算法准确定位了针对水印序列的攻击发生的位置，并报出了相应的异常详情。

```
解码失败！检测到非法编码！
水印值异常，异常行：1544，原因：水印无法被正确解码。
异常要素信息：
异常要素起始行：1544,要素类型：road,要素ID：10
```

(a)检测结果：一处水印序列不完整

```
水印值异常，异常行：1627，原因：水印值缺失。
异常要素信息：
异常要素起始行：1627,要素类型：junction,要素ID：42
```

(b)检测结果：一处水印序列缺失

```
水印值异常，异常行：1627，原因：树状水印内容异常。
水印值异常，异常行：1645，原因：树状水印内容异常。
异常要素信息：
异常要素起始行：1627,要素类型：junction,要素ID：42
异常要素起始行：1645,要素类型：junction,要素ID：43
```

(c)检测结果：两处水印序列异常

图 12 针对水印序列的攻击对应验证结果

Fig. 12 Validating Result of Attack on Watermark Sequence

2.3 水印导致高精地图文件大小变化分析

为了解使用本文算法嵌入水印后高精地图文件大小产生的变化，向多份高精地图文档示例数据中嵌入了水印，对比嵌入前后的文件大小，部分结果如表 4 所示。

表 4 嵌入水印后文件大小变化情况

Tab. 4 File Size Expansion after Embedding Watermark

文件名	要素数目	嵌入水印前大小 (Bytes)	嵌入水印后大小 (Bytes)	要素平均长度(Bytes)	增量与原始大小的比值
UC_ParamPoly3.xodr	11	37051	38363	3368.3	3.54%
map4.xodr	52	325224	333439	6254.3	2.53%
新元高速_20221013.xodr	314	939851	978380	2993.2	4.10%
changsha_part1_1014.xodr	434	2370133	2375950	5461.1	2.45%
Germany_2018.xodr	179	12594829	12682053	70362.2	0.69%

由表 4 可知，对于多份不同的高精地图数据，嵌入水印造成的增量在原始文件大小

的 0.69%~4.10%之间。描述每个要素的文本的平均长度越大，文件大小变化的程度越轻。通常情况下，描述每个要素的文本的平均长度不会低于 3000 字节，此时增量与原始大小的比值可以控制在 5%以内。因此，利用本算法嵌入水印不会导致高精地图文件大小剧烈变化。

3 总结与展望

本文针对 OpenDRIVE 格式的高精地图数据的完整性保护问题，选用不可见性良好的 Unicode 零宽度字符作为载体，利用前缀编码与信息-摘要算法，结合文档自身结构，提出了一种适用的脆弱水印算法。实验表明：本算法达到以下设计要求：

（1）向高精地图文档中嵌入水印后，不会造成肉眼可见的异常显示现象。

（2）嵌入水印后不会导致高精地图文件大小剧烈变化，增量与原始大小的比值可以控制在 5%以内。

（3）面对针对结点以及针对水印的攻击具有高度的敏感性，且能够准确定位篡改发生的位置。发生整要素删除时，能推断被删除要素的几何信息。

除此以外，对于其它结构相似的数据，如超文本标记语言（hyper-text markup language，HTML）文档，层叠样式表（cascading style sheet，CSS）文档，只要取消部分预处理要求、重新确定合适的水印嵌入位置并重新明确要素的定义以及要素间的排序规则，本算法就可以用于这些数据的完整性保护工作。

参 考 文 献

[1] Ministry of Transport of the People's Republic of China. The 14th Five Year Plan for the Development of Digital Transportation. （中华人民共和国交通运输部.数字交通“十四五”发展规划[EB/OL].(2021-10-25)[2023-12-5].https://www.mot.gov.cn/zhuanti/shisiwujtysfzgh/20220 1/P020220112576470472593.pdf）

[2] LI Bijun,GUO Yuan,ZHOU Jian,et al.Development and Prospects of High Definition Map for Intelligent Vehicle[J]. Geomatics and Information Science of Wuhan University,2024, 49(4):491-505.（李必军，郭圆，周剑，等.智能驾驶高精地图发展与展望[J].武汉大学学报（信息科学版）,2024, 49(4):491-505.）

[3] QI Ruyu, YIN Zhangcai, GU Jiangyan, et al. Knowledge Graph Expression of High Definition Map[J]. Geomatics and Information Science of Wuhan University,2024,49(4):651-661.（齐如煜，尹章才，顾江岩，等.高精地图的知识图谱表达[J].武汉大学学报（信息科学版）, 2024, 49(4): 651-661.）

[4] Wang Xiaoyun, Yu Hongbo. Survey of Hash Functions [J]. Journal of Information Security Research, 2015, 1(1):19-30.（王小云，于红波.密码杂凑算法综述[J].信息安全研究，2015, 1(1):19-30.）

[5] Zhu Changqing, Ren Na, Xu Dingjie. Geo-information security technology: progress and prospects[J]. 2022, 51(6): 1017-1028.（朱长青，任娜，徐鼎捷.地理信息安全技术研究进展与展望[J].测绘学报,2022,51 (6):1017-1028.）

[6] Hou Xiang, Min Lianquan, Tang Liwen. Fragile Watermarking Algorithm for Locating Tampered Entity Groups in Vector Map Data[J]. Geomatics and Information Science of Wuhan University, 2020, 45(2): 309-316.（侯翔， 闵连权， 唐立文. 定位篡改实体组的矢量地图脆弱水印算法[J]. 武汉大学学报(信息科学版), 2020, 45(2): 309-316.）

[7] Zhan Jiao, Guo Chi, Lei Tingting, et al. 2021. Comparative study on data standards of autonomous driving map[J]. Journal of Image and Graphics, 26(1): 0036-0048（詹骄，郭迟，雷婷婷,等. 自动驾驶地图的数据标准比较研究[J]. 中国图象图形学报, 2021,26(1) : 0036-0048）

[8] Zhao Weijuan, Guan Hu, Huang Ying, et al. A survey of text watermarking[J]. Journal of Communication University Of China (Science And Technology), 2020, 27(6):55-62.（赵卫娟，关虎，黄樱,等.文本水印技术研究综述[J].中国传媒大学学报(自然科学版),2020, 27(6):55-62.）

[9] Wang Bingxi, Chen Qi, Deng Fengsen. Technology of Digital Watermarking[M]. XiDian University Press, 2003.（王炳锡，陈琦，邓峰森.数字水印技术[M].西安电子科技大学出版社,2003.）

[10] Li Q, Zhang J, Zhang Z, et al. 2008. A Chinese Text Watermarking Based on Statistic of Phrase Frequency[C]. The 2008 International Confer-ence on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '08). IEEE Computer Society, USA, 2008, 335–338.

[11] Mir N, Khan M.A.U. Copyright Protection for Online Text Information: Using Watermarking and Cryptography[C]. 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2020, 1-4.

[12] Cai Yi, Li Feng, Zhou Liang, et al. An Digital Watermark Technology based on XML[J]. Modern Computer, 2005(6):66-68+83.（蔡毅，李峰，周亮等.一种基于 XML 的数字水印技术[J].现代计算机(专业版),2005(6):66-68+83.）

[13] Yao Ronghua, Zhao Qijun, Lu Hongtao. Algorithm for integrity protection for XML documents based on watermark[J].Computer Applications and Software, 2008(6):30-32.（姚荣华，赵启军，卢宏涛.基于水印的 XML 文档完整性保护算法[J].计算机应用与软件,2008(6):30-32.）

[14] Chen L, He W, Shu H, et al. Research on the Method of Text Information Hiding Based on XML[J]. Applied Mechanics & Materials, 2013, 385-386:1665-1668.

[15] Si Qin, Zhang Li, Lian Deliang. Text watermarking based on text feature[J].Journal of Computer Applications,2009,29(9):2348-2350.（斯琴，张力，廉德亮.基于文本特征的文本水印算法[J].计算机应用,2009,29(9):2348-2350.）

[16] Al-Wesabi F N, Alrowais F, Mohamed H G, et.al. Heuristic Optimization Algorithm Based Watermarking on Content Authentication and Tampering Detect-ion for English Text[J]. IEEE Access, Volume 11,2023:86104-86111.

[17] Li Zhaocan, Wang Liming, Ge Sijiang, et al. Big Data Plain Text Watermarking Based on Orthogonal Coding[J]. Computer Science,2019,46(12):148-154.（李兆璨，王利明，葛思江,等.基于正交编码的大数据纯文本水印方法[J].计算机科学,2019, 46(12):148-154.）

[18] Chen Yini, Li Qianmu, Lyu Chaoxian, et al. Research on text security hiding algorithm for invisible characters[J]. Cyberspace Security,2019,10(5):88-96.（陈旖旎， 李千目，吕超贤,等.不可见字符的文本安全隐藏算法研究[J].网络空间安全,2019,10(5):88-96.）

[19] Peng Deng. Webpage Link Tamper Localization Algorithm Based on Control Character Encoding[D]. Southwest Jiaotong University, 2016.（彭登. 基于控制符编码的网页链接篡改定位算法[D].西南交通大学,2016.）

[20] Xiao C, Zhang C, Zheng C. Fontcode: Embedding information in text documents using glyph perturbation[J]. ACM Transactions on Graphics (TOG), 2018, 37(2): 15.

[21] Qi W, Guo W, Zhang T, et al. Robust authentication for paper-based text documents based on text watermarking technology[J]. Mathematical Biosciences and Engineering, 2019, 16(4): 2233–2249.

[22] Sun Shan, Zhang Weiming, Fang Han, et al. Automatic generation of Chinese document watermarking fonts[J]. Journal of Image and Graphics, 2022, 27(1):0262-0276（孙杉，张卫明，方涵,等.中文水印字库的自动生成方法[J]. 中国图象图形学报, 2022, 27(1): 262－276.）

[23] Yao Ye, Liu Shuhui, Wang Hui, et al. Robust Chinese text watermarking method based on Chinese character glyph perturbation and font replacing[J]. Journal of Cryptologic Research, 2023, 10(4): 769–785. （姚晔，刘书辉，王慧,等.基于字符扰动变形和字库替换的鲁棒中文文本水印[J]. 密码学报, 2023, 10(4): 769－785.）

网络首发：
标题：面向 OpenDRIVE 格式高精地图的脆弱水印法
作者：袁天洋，朱长青，陈会仙，任娜，吕旭超
收稿日期：2024-04-18
DOI:10.13203/j.whugis20230500

引用格式：
袁天洋, 朱长青, 陈会仙,等. 面向 OpenDRIVE 格式高精地图的脆弱水印法[J].武汉大学学报（信息科学版），2024，DOI: 10.13203/j.whugis20230500（YUAN Tianyang, ZHU Changqing, CHEN Huixian, et al. Fragile Watermarking Algorithm for High Precision Map of OpenDRIVE Format[J].Geomatics and Information Science of Wuhan University, 2024 , DOI: 10.13203/j.whugis20230500）