



(12) 发明专利申请

(10) 申请公布号 CN 104821169 A

(43) 申请公布日 2015.08.05

(21) 申请号 201510228165.9

(22) 申请日 2013.04.08

(62) 分案原申请数据

201310119750.6 2013.04.08

(71) 申请人 宁波大学

地址 315211 浙江省宁波市江北区风华路
818 号

(72) 发明人 王让定 严迪群 金超 羊开云

(74) 专利代理机构 宁波奥圣专利代理事务所
(普通合伙) 33226

代理人 周珏

(51) Int. Cl.

G10L 25/78(2013.01)

G10L 19/018(2013.01)

权利要求书2页 说明书9页

(54) 发明名称

一种针对 MP3Stegz 的隐写检测方法

(57) 摘要

本发明公开了一种针对 MP3Stegz 的隐写检测方法,其先以二进制流方式打开待检测的 MP3 音频文件,然后按序从该 MP3 音频文件的二进制流中找到每一帧,接着通过分析每一帧中的第 37~41 个字节及第 42~56 个字节,确定每一帧是否经 MP3Stegz 隐写过,优点是通过分析该 MP3 音频文件的二进制流中的每帧中的第 37~41 个字节及第 42~56 个字节,能够准确地检测出该 MP3 音频文件是否经 MP3Stegz 隐写过,且性能稳健、检测准确率高、计算复杂度低。

1. 一种针对 MP3Stegz 的隐写检测方法,其特征在于包括以下步骤:

1) 以二进制流方式打开待检测的 MP3 音频文件;

2) 从该 MP3 音频文件的二进制流中找出第一帧,并将找出的第一帧定义为当前帧,再令 count 表示隐写的次序, count 的初始值为 0;

3) 判断当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数是否均为字符“X”的 ASC II 码,且当前帧中的第 41 个字节的 8 位二进制数是否为字符“1”~“9”中的任一个字符的 ASC II 码,如果是,则执行步骤 4), 否则,执行步骤 6);

4) 判断当前帧中的第 42 ~ 56 个字节中是否存在连续的两个字节满足以下条件:连续的两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续的两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码,如果存在,则令 count = count+1,其中,此处“=”为赋值符号,然后执行步骤 5), 否则,执行步骤 6);

5) 输出隐写的次序 count、隐写的起始位置,并提取出隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小,然后执行步骤 6);

6) 判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧,如果是,则执行步骤 7), 否则,从该 MP3 音频文件的二进制流中找出下一帧,将该帧作为当前帧,然后返回步骤 3) 继续执行;

7) 判断 count 的值是否为 0,如果是,则确定该 MP3 音频文件未经 MP3Stegz 隐写过,完成隐写检测,否则,确定该 MP3 音频文件经 MP3Stegz 隐写过,完成隐写检测。

2. 根据权利要求 1 所述的一种针对 MP3Stegz 的隐写检测方法,其特征在于所述的步骤 2) 中从该 MP3 音频文件的二进制流中找出第一帧的过程为:从该 MP3 音频文件的二进制流中的第 1 个字节开始,逐字节查询,当检索到连续的两个字节各自的 8 位二进制数依次为 11111111 和 1111101*时,确定这连续的两个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的第一帧开始的字节,其中,* 表示二进制数 0 或 1。

3. 根据权利要求 1 或 2 所述的一种针对 MP3Stegz 的隐写检测方法,其特征在于所述的步骤 6) 中从该 MP3 音频文件的二进制流中找出下一帧的过程为:根据当前帧的帧头内的信息计算出当前帧的长度,从当前帧的最后一个字节之后的一个字节开始,逐字节查询,当检索到连续的三个字节各自的 8 位二进制数依次满足:为 11111111、为 111*****、高 4 位不全为 1,则确定这连续的三个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的下一帧开始的字节,其中,* 表示二进制数 0 或 1。

4. 根据权利要求 3 所述的一种针对 MP3Stegz 的隐写检测方法,其特征在于所述的步骤 5) 中隐写的起始位置为当前帧中的第 41 个字节的位置。

5. 根据权利要求 4 所述的一种针对 MP3Stegz 的隐写检测方法,其特征在于所述的步骤 5) 中隐写的隐秘信息的格式由当前帧中的第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 2 个字节之后的三个字节确定:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。

6. 根据权利要求 5 所述的一种针对 MP3Stegz 的隐写检测方法,其特征在于所述的步骤 5) 中隐写的隐秘信息加密后的大小由当前帧中的第 40 个字节,与第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 1 个字节之间的若干字节确定:连续两个字节中的

第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。

一种针对 MP3Stegz 的隐写检测方法

[0001] 本申请是原申请号为 201310119750.6 的发明专利申请的分案申请,其申请日为 2013 年 04 月 08 日,发明名称为“一种针对 MP3Stegz 的隐写检测方法”。

技术领域

[0002] 本发明涉及一种音频隐写检测技术,尤其是涉及一种针对 MP3Stegz 的隐写检测方法。

背景技术

[0003] 隐写术是信息隐藏的一个重要分支,它是一种将隐秘信息隐藏在宿主信号中且在第三方不知道其存在的情况下通过公共信道传播的技术。隐写检测技术则是隐写术的对立技术,其目的是揭示可疑载体信号中是否存在隐秘信息以致破坏隐蔽通信。

[0004] 作为目前互联网上最流行的音频格式之一,MP3 音频得到了广泛的应用,因此针对 MP3 音频的隐写术和隐写检测技术也在博弈中不断发展。近年来出现了一些 MP3 音频的隐写工具,如 MP3Stego、UnderMP3Cover、MP3Stegz 等,与此同时针对各种隐写工具的隐写检测方法也被相继提出,如 Qiao 等人提出了一种针对 MP3Stego 的隐写检测方法、Jin 等人提出了一种针对 UnderMP3Cover 的隐写检测方法等。然而,目前针对 MP3Stegz 隐写工具的隐写检测方法还未见具体报道。

发明内容

[0005] 本发明所要解决的技术问题是提供一种针对 MP3Stegz 的隐写检测方法,其能够准确地检测出 MP3 音频是否被 MP3Stegz 隐写过,并且检测过程简单、计算复杂度低。

[0006] 本发明解决上述技术问题所采用的技术方案为:一种针对 MP3Stegz 的隐写检测方法,其特征在于包括以下步骤:

[0007] ①以二进制流方式打开待检测的 MP3 音频文件;

[0008] ②从该 MP3 音频文件的二进制流中找出第一帧,并将找出的第一帧定义为当前帧;

[0009] ③判断当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数是否均为字符“X”的 ASC II 码,且当前帧中的第 41 个字节的 8 位二进制数是否为字符“1”~“9”中的一个字符的 ASC II 码,如果是,则执行步骤④,否则,执行步骤⑤;

[0010] ④判断当前帧中的第 42 ~ 56 个字节中是否存在连续的两个字节满足以下条件:连续的两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续的两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码,如果存在,则确定该 MP3 音频文件经 MP3Stegz 隐写过,完成隐写检测,否则,执行步骤⑤;

[0011] ⑤判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧,如果是,则确定该 MP3 音频文件未经 MP3Stegz 隐写过,完成隐写检测,否则,从该 MP3 音频文件的二进制流中找出下一帧,将该帧作为当前帧,然后返回步骤③继续执行。

[0012] 所述的步骤②中从该 MP3 音频文件的二进制流中找出第一帧的过程为：从该 MP3 音频文件的二进制流中的第 1 个字节开始，逐字节查询，当检索到连续的两个字节各自的 8 位二进制数依次为 1111 1111 和 1111 101* 时，确定这连续的两个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的第一帧开始的字节，其中，* 表示二进制数 0 或 1。

[0013] 所述的步骤⑤中从该 MP3 音频文件的二进制流中找出下一帧的过程为：根据当前帧的帧头内的信息计算出当前帧的长度，从当前帧的最后一个字节之后的一个字节开始，逐字节查询，当检索到连续的三个字节各自的 8 位二进制数依次满足：为 1111 1111、为 111* ****、高 4 位不全为 1，则确定这连续的三个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的下一帧开始的字节，其中，* 表示二进制数 0 或 1。

[0014] 一种针对 MP3Stegz 的隐写检测方法，其特征在于包括以下步骤：

[0015] 1) 以二进制流方式打开待检测的 MP3 音频文件；

[0016] 2) 从该 MP3 音频文件的二进制流中找出第一帧，并将找出的第一帧定义为当前帧，再令 count 表示隐写的次序，count 的初始值为 0；

[0017] 3) 判断当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数是否均为字符“X”的 ASC II 码，且当前帧中的第 41 个字节的 8 位二进制数是否为字符“1”~“9”中的任一字符的 ASC II 码，如果是，则执行步骤 4)，否则，执行步骤 6)；

[0018] 4) 判断当前帧中的第 42 ~ 56 个字节中是否存在连续的两个字节满足以下条件：连续的两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码，且连续的两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码，如果存在，则令 count = count+1，其中，此处“=”为赋值符号，然后执行步骤 5)，否则，执行步骤 6)；

[0019] 5) 输出隐写的次序 count、隐写的起始位置，并提取出隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小，然后执行步骤 6)；

[0020] 6) 判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧，如果是，则执行步骤 7)，否则，从该 MP3 音频文件的二进制流中找出下一帧，将该帧作为当前帧，然后返回步骤 3) 继续执行；

[0021] 7) 判断 count 的值是否为 0，如果是，则确定该 MP3 音频文件未经 MP3Stegz 隐写过，完成隐写检测，否则，确定该 MP3 音频文件经 MP3Stegz 隐写过，完成隐写检测。

[0022] 所述的步骤 2) 中从该 MP3 音频文件的二进制流中找出第一帧的过程为：从该 MP3 音频文件的二进制流中的第 1 个字节开始，逐字节查询，当检索到连续的两个字节各自的 8 位二进制数依次为 1111 1111 和 1111 101* 时，确定这连续的两个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的第一帧开始的字节，其中，* 表示二进制数 0 或 1。

[0023] 所述的步骤 6) 中从该 MP3 音频文件的二进制流中找出下一帧的过程为：根据当前帧的帧头内的信息计算出当前帧的长度，从当前帧的最后一个字节之后的一个字节开始，逐字节查询，当检索到连续的三个字节各自的 8 位二进制数依次满足：为 1111 1111、为 111* ****、高 4 位不全为 1，则确定这连续的三个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的下一帧开始的字节，其中，* 表示二进制数 0 或 1。

[0024] 所述的步骤 5) 中隐写的起始位置为当前帧中的第 41 个字节的位置。

[0025] 所述的步骤 5) 中隐写的隐秘信息的格式由当前帧中的第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 2 个字节之后的三个字节确定：连续两个字节中的第 1 个

字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。

[0026] 所述的步骤 5) 中隐写的隐秘信息加密后的大小由当前帧中的第 40 个字节,与第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 1 个字节之间的若干个字节确定:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。

[0027] 与现有技术相比,本发明的优点在于:

[0028] 1) 通过分析 MP3 音频文件的每帧中的第 37 ~ 41 个字节及第 42 ~ 56 个字节,能够准确地检测出该 MP3 音频文件是否经 MP3Stegz 隐写过,且性能稳健、检测准确率高、计算复杂度低。

[0029] 2) 通过分析 MP3 音频文件中被 MP3Stegz 隐写过的帧中的第 41 ~ 56 个字节,找到连续的两个字节满足以下条件:前一个字节的 8 位二进制数为字符“#”的 ASC II 码,后一个字节的 8 位二进制数为字符“.”的 ASC II 码;然后根据此帧的第 40 个字节到这满足条件的连续的两个字节的第 1 个字节之间的若干个字节,能够获得隐秘信息加密后的大小,根据这满足条件的连续的两个字节中的第 2 个字节之后连续的三个字节能获得隐秘信息的格式,通过分析 MP3 音频文件中被 MP3Stegz 隐写过的帧中的第 41 ~ 56 个字节,能够获得当次隐写的起始位置。

具体实施方式

[0030] 以下结合实施例对本发明作进一步详细描述。

[0031] MP3Stegz 是目前流行的 MP3 隐写工具之一,它的隐写过程为:用由隐秘信息加密后转换成的字节替换可嵌帧的部分字节,同时它在隐写隐秘信息本身内容之前,会将隐秘信息的扩展名和加密后的大小隐写到 MP3 中。其整个隐写过程大致可分为两个部分:处理隐秘信息、隐写隐秘信息本身及其相关信息,处理隐秘信息部分主要完成:根据密码对隐秘信息进行加密,并提取扩展名和加密后的大小;隐写隐秘信息本身及其相关信息部分主要是根据隐写方法对前一部分提取的扩展名和大小以及隐秘信息的本身进行隐写。

[0032] 针对 MP3Stegz 的隐写原理,本发明提出了检测准确率高、计算复杂度低的针对 MP3Stegz 的隐写检测方法。

[0033] 实施例一:

[0034] 本实施例的隐写检测方法包括以下步骤:

[0035] ①以二进制流方式打开待检测的 MP3 音频文件。

[0036] ②从该 MP3 音频文件的二进制流中找出第一帧,并将找出的第一帧定义为当前帧。

[0037] 在此具体实施例中,步骤②中从该 MP3 音频文件的二进制流中找出第一帧的过程为:从该 MP3 音频文件的二进制流中的第 1 个字节开始,逐字节查询,当检索到连续的两个字节各自的 8 位二进制数依次为 1111 1111 和 1111 101* 时,确定这连续的两个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的第一帧开始的字节,其中,* 表示二进制数 0 或 1,即 1111 101* 包括 1111 1010 和 1111 1011。

[0038] ③判断当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数是否均为字符

“X”的 ASC II 码,且当前帧中的第 41 个字节的 8 位二进制数是否为字符“1”~“9”中的任一个字符的 ASC II 码,如果是,则执行步骤④,否则,执行步骤⑤。

[0039] ④判断当前帧中的第 42 ~ 56 个字节中是否存在连续的两个字节满足以下条件:连续的两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续的两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码,如果存在,则确定该 MP3 音频文件经 MP3Stegz 隐写过,完成隐写检测,否则,执行步骤⑤。

[0040] ⑤判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧,如果是,则确定该 MP3 音频文件未经 MP3Stegz 隐写过,完成隐写检测,否则,从该 MP3 音频文件的二进制流中找出下一帧,将该帧作为当前帧,然后返回步骤③继续执行。在此,判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧采用常见的技术,如当当前帧中的最后一个字节之后已无数据可供读入时,可确定该当前帧即为该 MP3 音频文件的二进制流中的最后一帧。

[0041] 在此具体实施例中,步骤⑤中从该 MP3 音频文件的二进制流中找出下一帧的过程为:根据当前帧的帧头内的信息(layer、bitrateIndex、samplingFrequency、paddingBit)计算出当前帧的长度,从当前帧的最后一个字节之后的一个字节开始,逐字节查询,当检索到连续的三个字节各自的 8 位二进制数依次满足:为 1 111 1111、为 111* ****、高 4 位不全为 1(对低 4 位不作要求),则确定这连续的三个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的下一帧开始的字节,其中,* 表示二进制数 0 或 1。

[0042] 实施例二:

[0043] 本实施例的隐写检测方法包括以下步骤:

[0044] 1) 以二进制流方式打开待检测的 MP3 音频文件。

[0045] 2) 从该 MP3 音频文件的二进制流中找出第一帧,并将找出的第一帧定义为当前帧,再令 count 表示隐写的次序,count 的初始值为 0。

[0046] 在此具体实施例中,步骤 2) 中从该 MP3 音频文件的二进制流中找出第一帧的过程为:从该 MP3 音频文件的二进制流中的第 1 个字节开始,逐字节查询,当检索到连续的两个字节各自的 8 位二进制数依次为 1111 1111 和 1111 101* 时,确定这连续的两个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的第一帧开始的字节,其中,* 表示二进制数 0 或 1,即 1111 101* 包括 1111 1010 和 1111 1011。

[0047] 3) 判断当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数是否均为字符“X”的 ASC II 码,且当前帧中的第 41 个字节的 8 位二进制数是否为字符“1”~“9”中的任一个字符的 ASC II 码,如果是,则执行步骤 4),否则,确定当前帧未经 MP3Stegz 隐写过,然后执行步骤 6)。

[0048] 4) 判断当前帧中的第 42 ~ 56 个字节中是否存在连续的两个字节满足以下条件:连续的两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续的两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码,如果存在,则令 count = count+1,其中,此处“=”为赋值符号,然后执行步骤 5),否则,执行步骤 6)。

[0049] 5) 输出隐写的次序 count、隐写的起始位置,并提取出隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小,然后执行步骤 6)。

[0050] 在此,隐写的起始位置为当前帧中的第 41 个字节的位置,即为当前帧中的第 37 ~ 40 个字节中的每个字节的 8 位二进制数均为字符“X”的 ASC II 码的最后一个字节之后的

一个字节。

[0051] 在此,隐写的隐秘信息的格式由当前帧中的第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 2 个字节之后的三个字节确定:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。假设当前帧中的第 45 个字节的 8 位二进制数为 0010 0011,是字符“#”的 ASC II 码,当前帧中的第 46 个字节的 8 位二进制数为 0010 1110,是字符“.”的 ASC II 码,则隐写的隐秘信息的格式由当前帧中的第 46 个字节之后的三个字节确定,即由当前帧中的第 47 个字节、第 48 个字节和第 49 个字节确定。

[0052] 在此,隐写的隐秘信息加密后的大小由当前帧中的第 40 个字节,与第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 1 个字节之间的若干字节确定:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。假设当前帧中的第 45 个字节的 8 位二进制数为 0010 0011,是字符“#”的 ASC II 码,当前帧中的第 46 个字节的 8 位二进制数为 0010 1110,是字符“.”的 ASC II 码,则隐写的隐秘信息加密后的大小由当前帧中的第 40 个字节与当前帧中的第 45 个字节之间的所有字节确定,即由当前帧中的第 41 ~ 44 个字节确定。

[0053] 6) 判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧,如果是,则执行步骤 7),否则,从该 MP3 音频文件的二进制流中找出下一帧,将该帧作为当前帧,然后返回步骤 3) 继续执行。在此,判断当前帧是否为该 MP3 音频文件的二进制流中的最后一帧采用常见的技术,如当前帧中的最后一个字节之后已无数据可供读入时,可确定该当前帧即为该 MP3 音频文件的二进制流中的最后一帧。

[0054] 在此具体实施例中,步骤 6) 中从该 MP3 音频文件的二进制流中找出下一帧的过程为:根据当前帧的帧头内的信息(layer、bitrateIndex、samplingFrequency、paddingBit)计算出当前帧的长度,从当前帧的最后一个字节之后的一个字节开始,逐字节查询,当检索到连续的三个字节各自的 8 位二进制数依次满足:为 1111 1111、为 111* ****、高 4 位不全为 1(对低 4 位不作要求),则确定这连续的三个字节中的第 1 个字节为该 MP3 音频文件的二进制流中的下一帧开始的字节,其中,* 表示二进制数 0 或 1。

[0055] 7) 判断 count 的值是否为 0,如果是,则确定该 MP3 音频文件未经 MP3Stegz 隐写过,完成隐写检测,否则,确定该 MP3 音频文件经 MP3Stegz 隐写过,完成隐写检测。

[0056] 上述实施例一仅检测了 MP3 音频文件是否经 MP3Stegz 隐写过,而实施例二在实施例一的基础上,再输出了每次隐写的次序 count、隐写的起始位置,并提取出了隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小。

[0057] 为更好地说明本发明方法的有效性和可行性,利用实施例二给出的隐写检测方法进行隐写检测实验。

[0058] 隐写检测实验一:

[0059] 对未经 MP3Stegz 隐写过的 MP3 音频文件进行隐写检测的过程为:

[0060] 1、从因特网上随机下载一个 MP3 格式的音频文件(如 testcase.mp3),testcase.mp3 的重要参数如表 1 所列,然后以二进制方式打开 testcase.mp3。

[0061] 表 1 MP3 音频 testcase.mp3 的重要参数列表

[0062]

文件大小	3776512 个字节
持续时间	3 分 55 秒
采样频率	44. 1KHz
比特率	128Kbps

[0063] 2、从 testcase.mp3 的二进制流中的第 1 个字节开始,逐字节查询,当检索到第 1378 和 1379 字节时,由于它们各自的 8 位二进制数依次为 1111 1111、1111 101*,满足成为 testcase.mp3 的二进制流中的第一帧的条件,因此确定第 1378 个字节为 testcase.mp3 的二进制流中的第一帧开始的字节,然后将第一帧作为当前帧。

[0064] 3、检查当前帧中的第 37 ~ 41 个字节,发现前 4 个字节中的每个字节的 8 位二进制数均为 1111 1111,都不是字符“X”的 ASC II 码;同样发现第 5 个字节的 8 位二进制数为 1111 1111,不是字符“1”~“9”中任意一个字符的 ASC II 码,因此确定当前帧未经 MP3Stegz 隐写过。

[0065] 4、检查当前帧是否为 testcase.mp3 的二进制流中的最后一帧,当不是最后一帧时,根据当前帧的帧头内的信息计算得到当前帧的长度为 418 个字节,并获得当前帧的最后一个字节,为第 1796 个字节,然后从第 1797 个字节开始逐字节查询,当检索到连续的三个字节各自的 8 位二进制数依次满足:为 1111 1111、为 111* ****、高 4 位不全为 1(对低 4 位不作要求),则确定这连续的三个字节中的第 1 个字节为 testcase.mp3 的二进制流中的下一帧开始的字节。

[0066] 5、将下一帧作为当前帧,按上述过程 3 和 4 继续检测该当前帧是否经 MP3Stegz 隐写过,依照此方法不断地检测每一帧,直到 testcase.mp3 的二进制流中的最后一帧,在检测过程中发现 testcase.mp3 的二进制流中的所有帧均未同时满足以下两个条件:①每帧中的第 37 ~ 41 的 5 个字节中,前 4 个字节中的每个字节的 8 位二进制数均为字符“X”的 ASC II 码,第 5 个字节的二进制数为字符“1”~“9”中任意一个字符的 ASC II 码;②每帧中的第 42 ~ 56 个字节中有满足以下条件的连续两个字节:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。因此,确定该 MP3 音频文件未经 MP3Stegz 隐写过,至此完成隐写检测。

[0067] 隐写检测实验二:

[0068] 对经 MP3Stegz 隐写过的 MP3 音频文件进行隐写检测的过程为:

[0069] 1、假设此处待检测 MP3 音频文件为 testcase_steg.mp3,是实验一中的 testcase.mp3 经 MP3Stegz 隐写 4 次得到的,且这 4 次隐写的具体内容如表 2 所列。

[0070] 表 2 testcase_steg.mp3 音频文件被隐写的次序及每次隐写的内容的列表

隐写次序	隐写内容		
	类型	格式	加密后大小
[0071] 第 1 次	视频	FLV	43213 个字节
第 2 次	图片	JPG	7072 个字节
第 3 次	音频	MP3	16432 个字节
第 4 次	文本	TXT	52 个字节

[0072] 2、从 testcase_steg.mp3 的二进制流中的第 1 个字节开始,逐字节查询,当检索到第 1378 和 1379 字节时,由于它们各自的 8 位二进制数依次为 1111 1111、1111 101*,满足成为 testcase_steg.mp3 的二进制流中的第一帧的条件,因此确定第 1378 个字节为 testcase_steg.mp3 的二进制流中的第一帧开始的字节,然后将第一帧作为当前帧。

[0073] 3、检查当前帧中的第 37 ~ 41 个字节,发现前 4 个字节中的每个字节的 8 位二进制数均为 0101 1000,都是字符“X”的 ASC II 码,同样发现第 5 个字节的 8 位二进制数为 0011 0100,即为字符“4”的 ASC II 码;再检查当前帧中的第 42 ~ 56 个字节,发现第 46 个字节的 8 位二进制数为 0010 0011,是字符“#”的 ASC II 码,第 47 个字节的 8 位二进制数为 0010 1110,是字符“.”的 ASC II 码;此时使隐写的次序 count 加 1。

[0074] 4、输出隐写的次序 count 和隐写的起始位置,检测出该次隐写的次序为第 1 次;隐写的起始位置为当前帧中的第 41 个字节的位置,即为 testcase_steg.mp3 的二进制流中的第 1418 个字节。

[0075] 提取出隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小,在第 1 次隐写中,将当前帧中的第 40 个字节与第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 1 个字节(对于第一帧,即为第一帧中的第 46 个字节)之间的所有字节提取出来:连续两个字节中的第 1 个字节的 8 位二进制数为字符“#”的 ASC II 码,且连续两个字节中的第 2 个字节的 8 位二进制数为字符“.”的 ASC II 码。即提取出当前帧中的第 41 ~ 45 个字节,它们各自的 8 位二进制数分别为 0011 0100、0011 0011、0011 0010、0011 0001、0011 0011,其中,0011 0100 为字符“4”的 ASC II 码,0011 0011 为字符“3”的 ASC II 码,0011 0010 为字符“2”的 ASC II 码,0011 0001 为字符“1”的 ASC II 码,0011 0011 为字符“3”的 ASC II 码,即得到隐写的隐秘信息加密后的大小为 43213 个字节;在第 1 次隐写中,将当前帧中的第 42 ~ 56 个字节中满足以下条件的连续两个字节中的第 2 个字节(对于第一帧,即为第一帧中的第 47 个字节)之后的三个字节提取出来,即提取出第 48 个、第 49 个和第 50 个字节,它们各自的 8 位二进制数分别为 0110 0110、0110 1100、0111 0110,其中,0110 0110 为字符“f”的 ASC II 码,0110 1100 为字符“1”的 ASC II 码,0111 0110 为字符“v”的 ASC II 码,即表示隐写的隐秘信息的格式为 flv。

[0076] 5、检查当前帧是否为 testcase_steg.mp3 的二进制流中的最后一帧,当不是最后一帧时,根据当前帧的帧头内的信息计算得到当前帧的长度为 418 个字节,并获得当前帧的最后一个字节为 testcase.mp3 的二进制流中的第 1796 个字节,然后从第 1797 个字节开始逐字节查询,当检索到连续的三个字节中的每个字节的 8 位二进制数依次满足:为 1111 1111、为 111* ****、高 4 位不全为 1(对低 4 位不作要求),则确定这连续的三个字节中的第 1 个字节为 testcase_steg.mp3 的二进制流中的下一帧开始的字节。

[0077] 6、将下一帧作为当前帧,按上述过程3和4继续检测该当前帧是否经MP3Stegz隐写过,依照此方法不断地检测每一帧,直到testcase.mp3的二进制流中的最后一帧,在检测过程中发现testcase_steg.mp3的二进制流中的第8944帧、第8964帧、第9009帧均同时满足以下两个条件:①每帧中的第37~41的5个字节中,前4个字节中的每个字节的8位二进制数均为字符“X”的ASC II码,第5个字节的8位二进制数为字符“1”~“9”中任意一个字符的ASC II码;②每帧中的第42~56个字节中有满足以下条件的连续两个字节:连续两个字节中的第1个字节的8位二进制数为字符“#”的ASC II码,连续两个字节中的第2个字节的8位二进制数为字符“.”的ASC II码。对于第8944帧,检测出为隐写的次序为第2次,隐写的起始位置为testcase.mp3的二进制流中的第3739227个字节,隐写的隐秘信息的格式为jpg,隐写的隐秘信息加密后的大小为7072个字节;对于第8964帧,检测出为隐写的次序为第3次,隐写的起始位置为testcase.mp3的二进制流中的第3747586个字节,隐写的隐秘信息的格式为mp3,隐写的隐秘信息加密后的大小为16432个字节;对于第9009帧,检测出为隐写的次序为第4次,隐写的起始位置为testcase.mp3的二进制流中的第3766394个字节,隐写的隐秘信息的格式为txt,隐写的隐秘信息加密后的大小为52个字节。

[0078] 7、此时count的值为4,因此确定该MP3音频文件经MP3Stegz隐写过,且隐写的次数为4次,至此完成隐写检测。

[0079] 通过上述两个实验可以很好地说明本发明方法能够准确的检测出MP3音频文件是否经MP3Stegz隐写过,并能够准确的获得隐写的次序、隐写的起始位置、隐写的隐秘信息的格式和隐写的隐秘信息加密后的大小。这足以说明了本发明方法的有效性和可行性。

[0080] 以下为对本发明方法的性能进行分析。

[0081] 第一类样本为未经MP3Stegz隐写的MP3音频文件:200首因特网上随机下载的MP3格式的音频文件,每个MP3音频文件3~5分钟,7种编码器类型(包括Lame、Xing、Gogo、Blade、FhG、Shine、Plugger),3种编码模式(VBR(可变速率)、ABR(平均速率)以及CBR(固定速率)三种,其中CBR情况下,编码速率包括96、112、128、160、192、256、320kbps等);第二类样本为经MP3Stegz隐写的MP3音频文件:将格式分别为TXT、MP3、JPG、FLV的4种多媒体文件作为隐秘信息,并通过MP3Stegz随机隐写进第一类样本的200个样本中,最后产生分别隐写1次、2次、3次、4次的样本各50个,共200个。

[0082] 利用本发明方法对上述两类样本中的所有样本进行隐写检测,检测的虚警率、漏检率和正检率如表3所列。

[0083] 表3 利用本发明方法对未经MP3Stegz隐写的MP3音频文件和经MP3Stegz隐写的MP3音频文件进行隐写检测的结果

[0084]

测试样本	虚警率 (%)	漏检率 (%)	正检率 (%)
第一类样本和第二类样本	0	0	100
虚警率: 未经 MP3Stegz 隐写的 MP3 音频文件错误地检测为经隐写过的概率; 漏检率: 经 MP3Stegz 隐写的 MP3 音频文件错误地检测为未隐写过的概率; 正检率: $1 - (\text{虚警率} + \text{漏检率}) / 2$ 。			

[0085] 从表 3 中可以看出, 本发明方法能够 100% 正确地判断一首 MP3 格式的音频文件是否经过 MP3Stegz 的隐写操作。表 4 给出了利用本发明方法对上述两类样本中的所有样本进行隐写检测的平均耗时。

[0086] 表 4 利用本发明方法对上述两类样本中的所有样本进行隐写检测的平均耗时

测试样本	检测平均耗时 (毫秒/个)
第一类样本	345
第二类样本	361
[0087]	测试软硬件环境: 计算机型号: 联想 启天 M730E CPU: Intel Core i3, 3.19GHz 内存: 1.74GB
[0088]	操作系统: Windows XP, 32bits 集成开发环境: Microsoft Visual Studio 2010

[0089] 从表 4 中可以看出, 本发明方法的计算复杂度低, 运行效率较高, 平均检测一首正常大小的 MP3 音频文件 (3 ~ 5 分钟) 耗时在 0.3 秒到 0.4 秒之间, 性能优良。