

# javascript实现图片隐写

原创

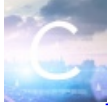
[yangnianbing110](#) 于 2016-07-15 18:28:26 发布 4128 收藏 3

分类专栏: [javascript](#) 文章标签: [javascript](#) [图片隐写](#) [canvas](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/yangnianbing110/article/details/51920161>

版权



[javascript](#) 专栏收录该内容

31 篇文章 0 订阅

订阅专栏

## 什么是图片隐写?



你在這兒不管怎麼找

发福利了，这是一张图片，这不仅仅是一张图片，如上图将一些信息隐藏在图片中的方式称之为图片隐写，但是有一点既然是隐写，那么肯定是悄悄的写，

图片还是原来的图片，只是里面隐藏了其他的信息。对于上面的图片，保存到本地之后，修改后缀名为.zip，然后用winrar打开，发现这是一个正常的压缩文件，里面包含了我们的福利信息。

这里讨论两种往图片内写入其他信息的方式，一种是在图片文件尾部添加信息，由于每种图片都有自己的文件编码格式，例如：对于png格式的图片，我们用winHex打开查看，发现文件末尾几个字节的数据总是固定的。

00001DA0	2F 1F DB A7 5D BD 21 10 3A 09 8A A2 F0 B2 C3 77	/ Ú\$]½! : !çð²Ãw
00001DB0	DF 7D 57 53 4B 2C 16 CB C4 C4 44 A1 50 F0 F9 7C	B}WSK, EÄÄDIPðù
00001DC0	A5 52 C9 EC 17 A5 01 3D 50 72 FE 99 C9 24 67 83	¥RÉi ¥ =Prp É\$g
00001DD0	07 7F 5E 10 88 4D 26 F4 B5 B1 69 5A C4 48 9B F1	^  M&ðµ±iZÄH ñ
00001DE0	AA 7B DB D8 D8 34 06 CD BA 09 D5 42 04 C2 5B 05	æ{Ú004 íº ÖB Ä[
00001DF0	8B C5 D2 D3 D3 33 30 30 30 31 31 61 BA 79 2F B4	!Ä000300011aºy/'
00001E00	4B 00 50 57 57 D7 CA 06 B3 43 27 7C 9A 1D B9 EB	K PwW×É °C'    'è
00001E10	DB 4B 7A 83 26 4C F3 B6 BD DC 8D 6F 6C 6C 0C 85	ÚKz!&Ló¶½Ú oll
00001E20	87 FF 5F C4 15 00 E0 C0 8D 30 01 BC 37 23 2C D8	!ÿ_Ä àÀ 0 ¼7#,#
00001E30	D3 CC C8 90 63 F8 0E B3 AD 28 81 F0 76 81 F7 24	ÓÏÈ cø º-( ðv +\$
00001E40	34 36 36 36 35 35 65 DA 25 8A 91 0D 42 08 21 44	466655eÚ% ' B !D
00001E50	D3 34 BE E8 3C 3B 09 84 AE 0E 45 51 14 45 E1 2D	Ó4æ<; !@ EQ Eá-
00001E60	2F 19 2D FD 1F A8 20 12 D6 3A 8F 70 69 00 00 00	/ -ý " Ö: pi
00001E70	00 49 45 4E 44 AE 42 60 82	IEND@B`

图片查看器查看图片的时候会忽略掉结束符之后的信息，所以不会影响到图片的显示。如果你用win系统，用copy命令就可以实现。

```
copy /b 1.png+2.rar 3.png
```

其中1.png是待写入信息的图片，2.zip是待写入的信息，3.png是最后得到的图片，修改3.png的名字为3.zip即可用压缩软件查看隐藏在图片中的信息。

还有另外一种方式是在人眼不可见的范围之内修改像素信息，将需要写入的信息按照一定的编码方式编码到图片的像素信息内，一种常见的方式就是最低有效位，就是将需要写入的信息

编码到图片像素点的最低有效位。我们知道每个像素都是由三原色红绿蓝三种颜色组成，一个像素点可以由4组8位二进制数表示，红绿蓝以及透明度，这里我们将信息编码到图片的透明度信息中，

透明度分为255级，如果只是修改8位2进制的最后一位，一般人的眼睛是识别不出来。

这里我们用javascript中的canvas来实现编辑图片的像素信息，并将一段文本编码到图片信息中。

```
function strToUint16Array(str){
    var buf = new ArrayBuffer(str.length * 2);
    var bufView = new Uint16Array(buf);
    for(var i = 0, len = str.length; i < len; i++){
        bufView[i] = str.charCodeAt(i);
    }
    return bufView;
}

var uint16 = strToUint16Array(msg);
var uint16Binary = [];
var msgBinary = uint16.forEach(function(data){
    data = data.toString(2);
    uint16Binary.push(appendStr(data+'', '0', 16));
});

uint16BinaryStr = uint16Binary.join('');
```

首先我们通过上面的函数将文本串每个字符的unicode码点信息存入到一个Uint16Array对象中，Uint16Array是ES6中一个新的数据类型，对javascript操作二进制数据提供支持。将Uint16Array中的元素码点都转换为二进制，长度不足16的前缀补0。

```
var image = new Image();
image.src = event.target.result;

var canvas = document.createElement('canvas');
canvas.height = image.height;
canvas.width = image.width;
document.querySelector('body').appendChild(canvas);
//document.querySelector('body').appendChild(image);
var ctx = canvas.getContext('2d');

ctx.drawImage(image, 0, 0);
var imageData = ctx.getImageData(0, 0, image.width, image.height);
```

接下来我们通过canvas的getImageData()来获取图片的像素信息。getImageData()方法返回一个ImageData对象，该对象有三个属性ImageData.data, ImageData.height, ImageData.width，其中ImageData.data类型是Uint8ClampedArray，其中每四个元素表示一个像素点的RGBA信息，接下来遍历像素点信息，写入之前文字编码之后的数据。

```
for(var i = 0; i < imageData.data.length; i++){
    var piexData = imageData.data[i];
    if(!((i+1) % 4)){
        var hexPiexData = piexData.toString(2);
        var changeValue = uint16BinaryStr.charAt((i+1)/4-1);
        if(changeValue){
            hexPiexData = hexPiexData.substring(0, hexPiexData.length-1) + changeValue;
        }
        imageData.data[i] = parseInt(hexPiexData,2);
    }else{
        imageData.data[i] = piexData;
    }
}
```

imageData.data的每第四位表示像素点的透明度，也是一个0-255的整数，也就是8位二进制，把文字编码后的二进制字符串每一位，写到代码像素透明度8位2进制的最后一位。

通过canvas的putImageData()方法将imageData写入到canvas中，右键另存为可以把编码之后的图片保存到本地，就得到了一张打上你印记的图片。

上面把信息写到了文件中，人眼不可见，怎么把信息取出来，采用上面的逆序进行操作即可。

```
var image = new Image();
image.src = event.target.result;

var canvas = document.createElement('canvas');
canvas.height = image.height;
canvas.width = image.width;
document.querySelector('body').appendChild(canvas);
var ctx = canvas.getContext('2d');

ctx.drawImage(image, 0, 0);
var imageData = ctx.getImageData(0, 0, image.width, image.height);
```

拿到图片的像素信息之后，遍历像素数组，取出每四个元素中的数值并转换为8位进制表示方式，取最后一位即可。

```
var binarytextInImg='';
imageData.data.forEach(function(item, index){
    if(!((index+1)%4)){
        var piexData = imageData.data[index].toString(2);
        var lastIndexStr = piexData.charAt(piexData.length-1);
        binarytextInImg +=  lastIndexStr
    }
});
```

binarytextInImg变量的每十六位都是一个字符的码点，通过String.fromCharCode即可转换为对应的字符。

```
function binaryStrToUnit16Array(binaryStr){
    var buf = new ArrayBuffer(parseInt(binaryStr.length / 8));
    var bufView = new Uint16Array(buf);
    var value = '';
    for(var i = 0, len = binaryStr.length; i < len; i++){
        value += binaryStr.charAt(i)
        if(!((i+1)%16)){
            bufView[(i+1)/16-1] = parseInt(value, 2);
            value = '';
        }
    }
    return bufView;
}
var text = uint16ArrayToStr(binaryStrToUnit16Array(binarytextInImg));
```

得到的text就是隐藏在图片中的文本信息，这个二进制倒腾来倒腾去的好麻烦啊！

写信息戳[这里](#)

读信息戳[这里](#)

更为详细的图片隐写技术介绍戳[这里](#)