#### 应用场景

我们在访问网站时，看到的URL可能会是这样的，如下：

http://www.projectd.com/BurnTaskLog.aspx?UserRoles=CommonUser&id=201002010201

场景一：

我们可以通过修改查询字符串的值来访问其他的页面，比如将id=201002010201修改为id=201002010202，这样可能得到其他的信息，如果每个页面都有一定的信息，那么我们可以根据这个id值的规律来制作一个程序从而获取全部页面的重要信息。

场景二：

如果这个页面是一个机密程序中的一个页面，这个页面中包含用户A的机密信息，用户B可能会拷贝这个URL值来查看用户A的信息，造成信息的泄露。

针对这两种情况，如何进行解决？

#### 解决方案

我们可以通过对问号后面的查询字符串进行加密，加密原理是先对查询字符串进行异或加密，然后对异或加密后的字符串进行base64加密。

异或加密的key如何选取？

对于场景一：这个key可以在程序中内定。

对于场景二：这个key可以是用户的id和一定字符串的混合，单独的用户id可能也会造成url破解。

不可行的方案：只是对查询字符串进行base64加密，这样的密文他人仍然可以通过base64再进行解密。通过混合异或加密算法，他人如果不知道异或的key，那么就不会解密成功。

#### 相关资料

[客户端base64加密解密](base64.js)

[服务器端base64加密解密](Base64.cs)

[客户端 xor 加密解密](xor.js)

[服务器端 xor 加密解密](XOR.cs)

#### 使用方法