常用链接 我的随笔

我的参与 最新评论

随笔分类

Android(70) Cloud(2) css(3) Delphi(2) Django(9) eclipse(1) English(1) Iphone(31) iPhone人机界面指南(11) iPhone应用程序编程指南(10) ipvs(lvs)(12) java(46)

javascript&jquery(9) Linux(151) Linux C(67) lua(1) maven(5) mongodb(8)

mysql(22) php(35) pvthon(92) redis(17) socket(2) Spring(16) Web(21) windbg(5) Windows(12)

WTL(2) 播客视频(6) 大数据小理(1) 多媒体编程(36) 个人日记(6) 个性化推荐与搜索(4) 管理(8) 汇编(4)

美食天下 其他(22) 设计模式(4)

数据结构(2) 算法&数据结构(1) 网络编程(11) 武(2) 系统编程(1)

系统架构(11) 协议(7) 形意拳(4) 研发质量管理(11) 音频(3) 转载日记(11)

自动化(6) 🌞 随笔档案

2016年1月 (5) 2015年12月 (3) 2015年11月 (2) 2015年10月 (1) 2015年9月(3) 2015年8月 (6) 2015年7月 (6) 2015年6月 (5) 2015年5月 (1) 2015年4月 (7) 2015年3月 (8) 2015年2月(3) 2015年1月 (9) 2014年12月 (6) 2014年11月 (13)

2014年10月 (16)

2014年9月 (9)

2014年8月 (4) 2014年7月 (1)

2014年6日 (14)

2014年4月 (9)

博客園 首页 新随笔 联系 管理 订阅 図 図

随笔 - 895 文章 - 0 评论 - 119

Base64加密原理(转)

Base64是一种加料置位加密法,那为什么叫base64呢?因为无论明文是什么(比如汉字,特殊符号等),加密后的密文都只会变成字母A-Z、a-z和0-9 和+和/这64个字符,被他加密体后积一般会变成原来的4/3。 Base64的标准在RFC2045里的24页可以看到。

http://wenku.baidu.com/view/bd567523aaea998fcc220e8e.html

下面我就来加密一个astralidea的简写AST来说明他的加密原理。

21

S 65 83 01010011 01000001 01010100 010000010101001101010100

为了说明方便,我们用人的思考方式,而不是计算机的方式,下面是上图的说明。

1.假设有3个字符, 首先先将他换成10进制ASCII码, 然后转成2进制, 少位用0补齐, 保证8位。

13

N

2.因为ASCII码大小是一字节也就是8位。3个8位。又因为3*8=4*6, 我们给他分成4分, 每份6位。如果少位都用0补齐。

20

U

3.把6位码前2位加两个0,比如010000转完后就是00010000,这对计算机来说比较重要,变成了8位码。

4.将4个8位(也就是一字节)码按照RFC里的BASE64标准换算, 下面也提供了标准表。

这样我们就得到了QVNU,也就是AST加密的结果。

下面是标准的BASE64编码表。

16

Q

Freed & Borenstein Standards Track [Page 24] RFC 2045 Internet Message Bodies November 1996

Table 1: The Base64 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding	
0	A	17	R	34	i	51	Z	
1	В	18	S	35	j	52	0	
2	C	19	T	36	k	53	1	
3	D	20	U	37	1	54	2	
4	E	21	V	38	m	55	3	
5	F	22	W	39	n	56	4	
6	G	23	X	40	0	57	5	
7	H	24	Y	41	p	58	6	
8	I	25	Z	42	a	59	7	
9	J	26	a	43	r	60	8	
10	K	27	b	44	3	61	9	
11	L	28	c	45	t	62	+	
12	M	29	d	46	u	63	1	
13	N	30	e	47	v			
14	0	31	f	48	w	(pad)	-	
15	P	32	q	49	x			
16	0	33	h	50	v			

那么有3N个字符就是分别把3个转成4个了, 那假如有3N+1个字符怎么办呢?

BASE64是这样规定的, 剩下的用"="补齐。

比如A这个字符转后为01000001末尾加4个0变为010000 | 010000查表得QQ差2个用=补齐。

因此A加密后是QQ==. 对于3N+2情况

同理AS加密后为QVM= 这样就有了所有情况了。

说了这么多,我们给出实现加密的代码,从这里更可以看出加密过程简洁的表达

```
part = (len+2) / 3;
newc = (unsigned char *)malloc(sizeof(unsigned char) * part * 4);
                       in[0] = \underline{s}[i*3], in[1] = \underline{s}[i*3+1], in[2] = \underline{s}[i*3+2];

add = \underline{len} - i*3;
                       newo[1*4] = charset[in[0] >> 2];

newo[1*4+1] = charset[(in[0] & 0x3) << 4) | ((in[1] & 0xf0) >> 4)];

newo[1*4+2] = (unsigned char) (add > 1 ? charset[((in[1] & 0xf) << 2)]

| ((in[2] & 0xc) >> 6) | 1 '='');

newo[1*4+3] = (unsigned char) (add > 2 ? charset[in[2] & 0x5f] : '=');
```

当然如果你用.net库你可以直接调用

byte[] bytes=Encoding.Default.GetBytes("ConvertString");
Convert.ToBase64String(bytes);

即可。

1.由于BASE64可以加密任何的字符,那么我们可以把一个小程序比如几K的EXE文件加密,然后就可以发到网页上,因为BASE64标准没有非法字符,放到任何空间都应该不会有问题。然后另一边复制下来解密就OK了,这样可以省得传到网络硬盘各种 原因(比如病毒上传失败)之类的问题。

2.BASE64最初是考虑加密后让你不能一眼看出明文是什么而出现的, 其最主要的应用当然是加密了, 比如迅雷就是通过BASE64加料后加密而实现的。

比如我虚拟一个地址http://www.astralidea.com/down.exe

我们知道这个地址根本不存在。假设我们会了BASE64怎么模拟迅雷的地址呢?

1.迅雷把上面地址加上料AA和ZZ变成: AAhttp://www.astralidea.com/down.exeZZ

2.把上面的字符加密得:QUFodHRwOi8vd3d3LmFzdHJhbGlkZWEuY29tL2Rvd24uZXhlWlo=

3.再前面加上thunder://得:thunder://QUFodHRwOi8vd3d3LmFzdHJhbGlkZWEuY29tL2Rvd24uZXhlWlo=

你会发现加密后的和我虚拟的是一样的。

FLASHGET原理也是一样,只不过加的料为[FLASHGET]。

而QQ旋风干脆不加料,直接就对地址进行BASE64编码了。

3.Asp.Net里的viewstate就是用BASE64加密所得的,如果了解了这个可以把他解密就能得到一个XML文件,找到key-value pairs



⊞ НЗ ВРМ 自开发 零实施的BPM 👀

- 「代對家」的手均短程和手均設等 写給未来的程序緩 高质量的工程代码为什么难写 循序渐进地代码重构 技术的正宗与野路子

- 2009年9月(3)

2008年6月 (3) 2008年5月 (1) 2008年4月(4) 2008年3月(2)

2014年2月 (2) 2014年1月(4) 2013年12月 (7) 2013年11月 (5)

2013年10月 (7) 2013年9月 (12)

2013年8月 (19)

2013年6月 (4) 2013年5月 (29) 2013年4月 (20)

2013年3月 (15)

2013年1月 (52) 2012年12月 (25) 2012年11月 (10)

2012年10月 (10)

2012年7月 (21) 2012年6月 (23) 2012年5月 (23)

2012年4月 (8) 2012年2月 (18) 2012年1月(3) 2011年12月 (38) 2011年11月 (19) 2011年9月 (9) 2011年8月 (9) 2011年7月 (17) 2011年6月 (25) 2011年4月 (2) 2011年3月 (4) 2011年2月 (6)

2011年1月 (6) 2010年11月 (13) 2010年10月 (11) 2010年10月(1 2010年8月 (22)

2010年6月(3)

2010年5月 (15) 2010年4月 (2) 2010年3月(6)

2010年1月(5) 2009年12月 (5) 2009年11月 (1)

2009年7月(2) 2009年6月 (9) 2009年3月(4) 2008年10月 (27)

卷编程 AstralWind

非常好的python博文

C语言常见问题集

c/c++博客(chio) ChinaUnix

CoderZh的技术博客

非常好的python文章, Python天 天美味系列

high scalability

HoNooD的Blog

有很多好的C++ on windows文

ian的个人博客

python、django、爬虫、android iTech--Python

很全的python参考博客

ITPUB Just Steps

linux c/c++、shell等编程

linux服务器 linux基础

Linux内核API手册(Kernel API)

Linux内核源码在线阅读(LXR) Linux设备驱动开发指南

NoSqlFan

redis, mongdb PHP手册

PHP资源分享门户 python学习blog

Qt - 实宁波 的专栏

gt博客 Steven.Leong的专栏

wuzhekai的专栏 点点星光(测试)

读书笔记 逛奔的蜗牛(c++/qt) 航空母舰

mysql

结构之法 算法之道 开源技术手册黄页

炼数成金 平凡的世界

深入分析Linux内核源码

小米饭团 一宁

mysql 自动化测试 走向架构师之路

最新评论

1. Re:Django URL传递参数的方 老司机啊 写了这么多

2. Re:Linux下的tar压缩解压缩命令

非常棒!看完之后我明白tar命令

的,现在要做一个质量体系的研发 管理!正如贴中所言。一个质量体系,最难做的就是研发管理了!希 望能够看到贴主的更多金玉良富! --qeo51215 4. Re:并发控制 mysql MylSAM表 识 讲得很好,收藏了。 --想听家乡话 5. Re:Python 使用 UTF-8 编码(转) 十分感谢!!! 6. Re:python怎样压缩和解压缩ZIP 文件(转) rar文件可以使用这个解压缩吗?谢 --咸鱼翻身 7. Re:Python模块之: configobj(转) 请问这个lib是否支持多个相同名字 的 section? 8. Re:了解负载均衡 会话保持 session同步(转) 利用cookie同步session。 下面写的缺点, 如果禁用COOKIE 会导致失效。 但是cookie禁用的话, sessionID 也不可以用了, 也就没有所谓的会 9. Re:@RequestMapping 用法详 解之地址映射(转) 学习了 10. Re:maven核心, pom.xml详解 (转) 学习。 --NewSea **阅读排行榜** 1. Linux下的tar压缩解压缩命令详 解(243795) 2. mysql Access denied for user ...,¬¬¬ Access denied for user root@localhost错误解决方法总结 (转)(192446) 3. @RequestMapping 用法评解之 地址映射(转)(141962) 4. linux常用命令大全(转)(106039) 5. 在SpringMVC中使用 @RequestBody注解处理Json时, 报出HTTP Status 415的解决方案 (82808) 6. iOS消息推送机制的实现 (76362) 7. 两种 js下载文件的方法(转) 8. @RequestBody, @ResponseBody 注解详解(转) (51112) 9. ulimit -c unlimited(48980) 评论排行榜 1. 用Android NDK编译FFmpeg(7) 2. linux常用命令大全(转)(7) 3. ipvs学习笔记(二)(4) 4. @RequestMapping 用法详解之 地址映射(转)(4) 5. iOS消息推送机制的实现(3) 6. 了解負载均衡 会话保持 session同步(转)(3) 7. 浅淡Redis数据库的键值设计 8. Android Linux下开发环境搭建 9. ARP欺骗的源代码(2) 10. 音频捕捉(directshow)(2) 推荐排行榜 1. Java内存泄露的理解与解决(转) (11) 2. Linux下的tar压缩解压缩命令详 4. @RequestMapping 用法详解之 地址映射(转)(7) 5. Java命名规范(6) 6. 了解负载均衡 会话保持 session同步(转)(5) 7. EAX、ECX、EDX、EBX寄存器的 作用(5) 8. iOS消息推送机制的实现(5) 9. Duplicate Elimination in Scrapy(转)(4) 10. maven核心, pom.xml详解(转) Copyright ©2017 一个人的天空@

3. Re:如何构建研发质量管理体 系? 感谢主贴分享!本人就是个管质量