

关键代码在

com.wuba.jiazheng.h.w (参数排序 加密算法输入)  
com.wuba.jiazheng.h.aa (参数排序中使用的比较函数)  
com.wuba.jiazheng.h.v (加密算法)  
com.wuba.jiazheng.b.a (添加 http 头 c)

由于代码经过混淆，所以一些名称不易读懂。

com.wuba.jiazheng.h.w (参数排序 加密算法输入)：

```
public static String a(BasicNameValuePair[] arg7, String arg8) {
    String v0;
    HashMap v2 = new HashMap();
    int v3 = arg7.length;//10
    int v1;
    for(v1 = 0; v1 < v3; ++v1) {
        ((Map)v2).put(arg7[v1].getName(), arg7[v1].getValue());
    }//v2 中存了多个个键值對(url 中的参数)

    ArrayList v3_1 = new ArrayList();
    if(arg7 == null || arg7.length == 0) {
        v0 = null;
        return v0;
    label_19:
        while(v1 < v4) {
            for(v0 = arg7[v1].getName(); v0.length() < 5; v0 = v0 + "`")
            {

                } //键值对中名字长度不足 5，则用`填充

            ((List)v3_1).add(v0);
            ++v1;
        }

        Collections.sort(((List)v3_1), new aa());//对键值中的键进行排序 排
序算法中也用到了加密算法！
        StringBuffer v1_1 = new StringBuffer();
        Iterator v3_2 = ((List)v3_1).iterator();
        while(v3_2.hasNext()) {
            v0 = v3_2.next().replaceAll("`", "");//去除刚才加上的`
            if(v1_1.toString().equals("")) {
                v1_1.append(v0).append("=").append(((Map)v2).get(v0));
            }
            else {
                v1_1.append("&").append(v0).append("=").append(((Map)v2).get(v0));
            }
        }
    }
```

```

        v1_1.append(arg8);

        v0 = v.a(v1_1.toString()); //调用加密算法 a 见下面
    }
    else {
        int v4 = arg7.length;
        v1 = 0;
        goto label_19;
    }

    return v0;
}

```

#####

比较算法，这里也用到了加密算法：

```

final class aa implements Comparator {
    aa() {
        super();
    }

    public int a(String arg3, String arg4) {
        return v.a(arg3).compareTo(v.a(arg4)); //v.a 即是加密算法
    }

    public int compare(Object arg2, Object arg3) {
        return this.a(((String) arg2), ((String) arg3));
    }
}

```

#####

com.wuba.jiazheng.h.v (加密算法)

```

private static char[] a;

    static {
        v.a = new char[] { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
'a', 'b', 'c', 'd', 'e',
        'f' };
    }

    public static String a(String arg9) {
        int v0 = 0;
        try {

```

```

        MessageDigest v1 = MessageDigest.getInstance("MD5");
        v1.update(arg9.getBytes("UTF8"));
        byte[] v2 = v1.digest();
        int v3 = v2.length;
        char[] v4 = new char[v3 * 2];
        int v1_1 = 0;
        while(v0 < v3) {
            int v5 = v2[v0];
            int v6 = v1_1 + 1;
            v4[v1_1] = v.a[v5 >>> 4 & 15];
            v1_1 = v6 + 1;
            v4[v6] = v.a[v5 & 15];
            ++v0;
        }

        String v0_3 = new String(v4);
        return v0_3;
    }
    catch(UnsupportedEncodingException v0_1) {
        v0_1.printStackTrace();
    }
    catch(NoSuchAlgorithmException v0_2) {
        v0_2.printStackTrace();
    }

    return null;
}

```

com.wuba.jiazheng.b.a (添加 http 头 c)

```

HttpGet v0_3 = this.c.a(ah.a("http://jzt.58.com/", arg9),
((NameValuePair[])v2));
v0_3.addHeader("i", "1");
v0_3.addHeader("c", w.a(v2, "~!@#$$%^&*"));
com.wuba.android.lib.a.a v0_4 = this.c.b(((HttpRequestBase)v0_3), new
b());

```

算法核心思想：打乱 url 的参数顺序并加密，服务器端重新使用该算法校验数据包

加密算法输入：

date=2014-12-15

13:00:00&lng=116.658112&cityid=1&page=1&type=1&pagesize=10&r=0.20097687607282344&address=GUJIN 古今 12&duration=3&lat=40.136514~!@#%^^&\*

url：

GET

/api/guest/workersnearby?r=0.20097687607282344&pagesize=10&duration=3&address=GUJIN%E5%8F%A4%E4%BB%8A12&page=1&lng=116.658112&type=1&date=2014-12-15+13%3A00%3A00&lat=40.136514&cityid=1 HTTP/1.1\r\n

c: 6c2db5c6f88c9caffb392c604e271231\r\n

#####

加密算法输入：

date=2014-12-14

16:35:12&lng=116.658112&cityid=1&type=1&r=0.46667775068186634&address=GUJIN 古今 12&duration=3.0&lat=40.136514~!@#%^^&\*

url：

GET

/api/guest/oneworkernearby?r=0.46667775068186634&duration=3.0&lng=116.658112&type=1&date=2014-12-14+16%3A35%3A12&address=GUJIN%E5%8F%A4%E4%BB%8A12&lat=40.136514&cityid=1 HTTP/1.1\r\n

c: cf8f70081ec036d74b5447cd0324b83f\r\n

#####

加密算法输入：

date=2014-12-14

16:37:52&lng=116.658112&cityid=1&type=1&r=0.6265088268210154&address=GUJIN 古今 12&duration=4.0&lat=40.136514~!@#%^^&\*

url：

GET

/api/guest/oneworkernearby?r=0.6265088268210154&duration=4.0&lng=116.658112&type=1&date=2014-12-14+16%3A37%3A52&address=GUJIN%E5%8F%A4%E4%BB%8A12&lat=40.136514&cityid=1 HTTP/1.1\r\n

e=2014-12-14+16%3A37%3A52&address=GUJIN%E5%8F%A4%E4%BB%8A12&lat=40.136514&cityid=1  
HTTP/1.1\r\n

c: 81b4f329d246ccdad4705de4f61b8100\r\n

#####