如何爬取i春秋网课

原创

~巴哥~ 于 2021-03-16 21:23:04 发布 1625 收藏 3

分类专栏： 爬虫 python 文章标签： python 爬虫

本文链接：https://blog.csdn.net/a854596855/article/details/114901485

爬虫 同时被 2 个专栏收录

9 篇文章 1 订阅

订阅专栏

python

7 篇文章 0 订阅

订阅专栏

单位购买了几个i春秋上的网络课程，用于业务培训，领导希望将这些课程爬取下来，可以离线观看，将这个任务交给我，经过一番努力，摸清了i春秋前端的视频解密的过程，实现了这个爬虫，现将整个过程记录下来。

i春秋(https://www.ichunqiu.com/)是国内一家知名的网络安全类媒体，上面有许多非常好的技术资料和视频课程。

环境配置

使用的环境

- windows 7 x64
- python3.7(Anaconda 3)
- vscode
- 火狐开发版
  需要使用的python包有
- requests 用于模拟http请求
- bs4 用于解析html文档
- pycryptodome 用于AES解密
  这些包可以使用清华的pip源进行安装

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pycryptodome
```

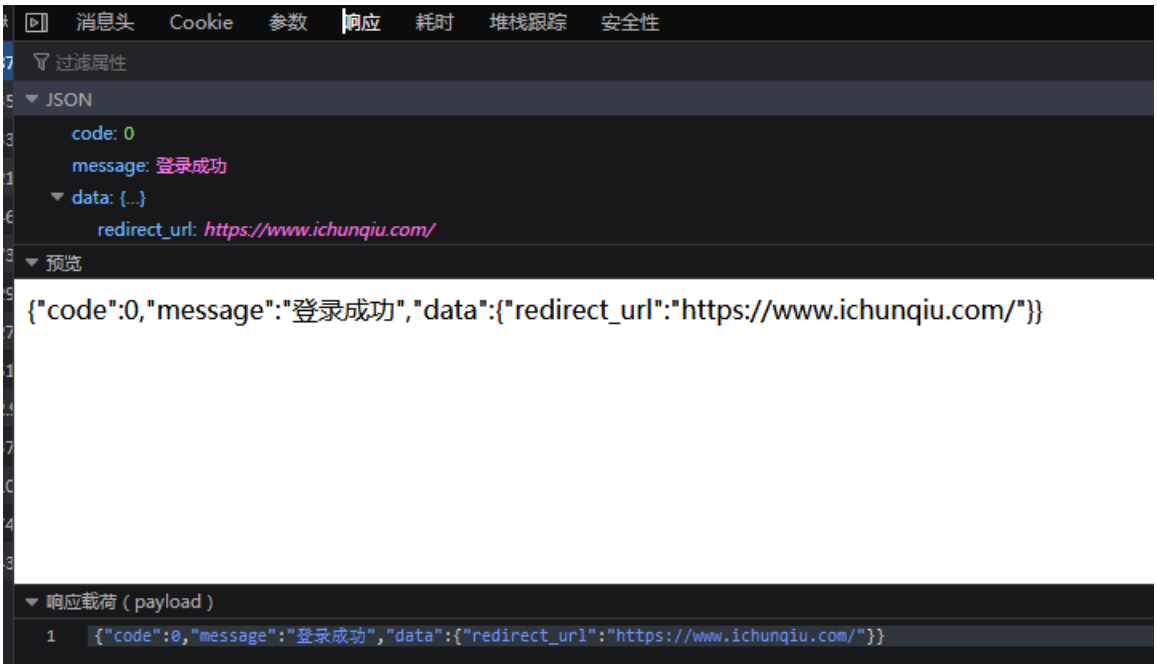
爬虫原理分析

网站登陆过程

使用火狐开发版，按F12，调出开发者工具，在网络选项卡中可查看网站前后端交互的所有HTTP请求和响应

登陆操作请求的地址为https://user.ichunqiu.com/login/normal
使用post方法，提交的参数为

返回的响应为



视频播放过程

登陆成功后，我们打开购买的视频
分析html结构，发现了页面中含有一个m3u8的地址

```
<div class="loadVideo" id="loadVideo" data-video-sourse="1" data-video-url="https://mv.ichunqiu.com/57765/57767/57769_d/57769.m3u8" data-video-resolution="0">
```

访问这个由m3u8地址,里面的内容为

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=10000000
/57765/57767/57769_d/720/57769.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=5000000
/57765/57767/57769_d/480/57769.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=2500000
/57765/57767/57769_d/320/57769.m3u8
```

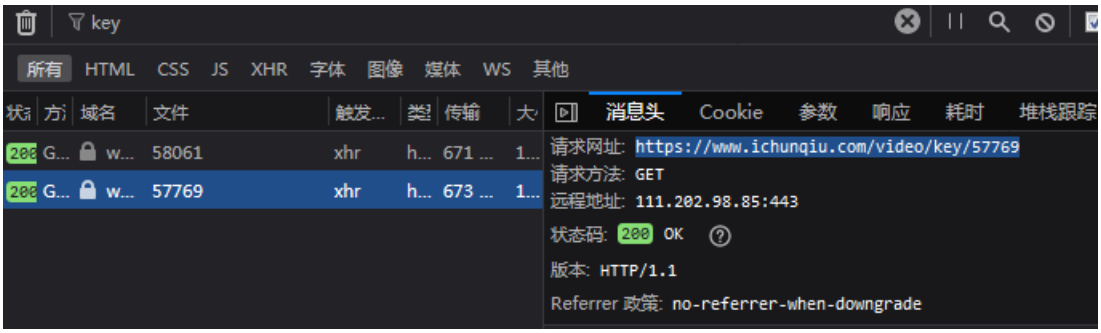
这是三种不同清晰度的视频的uri，以720p为例
完整的url为https://mv.ichunqiu.com/57765/57767/57769_d/320/57769.m3u8
查看其内容，发现这是一个标准的m3u8文件

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:29
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="http://www.ichunqiu.com/videokey?vid=57769",IV=0x99b74007b6254e4bd1c6e03631cad15b
#EXTINF:23.333333,
577690.ts
#EXTINF:20.833333,
577691.ts
#EXTINF:20.833333,
577692.ts
#EXTINF:20.250000,
577693.ts
#EXTINF:20.833333,
577694.ts
#EXTINF:20.833333,
577695.ts
#EXTINF:22.000000,
577696.ts
#EXTINF:20.833333,
577697.ts
#EXTINF:10.416667,
577698.ts
#EXTINF:20.833333,
577699.ts
#EXTINF:28.583333,
5776910.ts
#EXTINF:20.000000,
5776911.ts
#EXTINF:10.416667,
5776912.ts
#EXTINF:20.833333,
5776913.ts
#EXTINF:20.833333,
5776914.ts
#EXTINF:20.833333,
5776915.ts
#EXTINF:20.833333,
5776916.ts
#EXTINF:20.833333,
5776917.ts
#EXTINF:20.833333,
5776918.ts
#EXTINF:20.833333,
5776919.ts
```

#EXTINF:20.833333,
5776920.ts
#EXTINF:20.833333,
5776921.ts
#EXTINF:16.333333,
5776922.ts
#EXTINF:20.833333,
5776923.ts
#EXTINF:20.833333,
5776924.ts
#EXTINF:20.833333,
5776925.ts
#EXTINF:20.833333,
5776926.ts
#EXTINF:20.833333,
5776927.ts
#EXTINF:20.833333,
5776928.ts
#EXTINF:11.708333,
5776929.ts
#EXTINF:20.833333,
5776930.ts
#EXTINF:20.833333,
5776931.ts
#EXTINF:20.833333,
5776932.ts
#EXTINF:20.833333,
5776933.ts
#EXTINF:20.833333,
5776934.ts
#EXTINF:16.583333,
5776935.ts
#EXTINF:20.833333,
5776936.ts
#EXTINF:25.000000,
5776937.ts
#EXTINF:20.833333,
5776938.ts
#EXTINF:20.833333,
5776939.ts
#EXTINF:17.500000,
5776940.ts
#EXTINF:20.833333,
5776941.ts
#EXTINF:20.833333,
5776942.ts
#EXTINF:20.833333,
5776943.ts
#EXTINF:20.833333,
5776944.ts
#EXTINF:10.416667,
5776945.ts
#EXTINF:20.833333,
5776946.ts
#EXTINF:28.125000,
5776947.ts
#EXTINF:20.833333,
5776948.ts
#EXTINF:10.416667,
5776949.ts

```
#EXTINF:20.833333,
5776950.ts
#EXTINF:13.125000,
5776951.ts
#EXT-X-ENDLIST
```

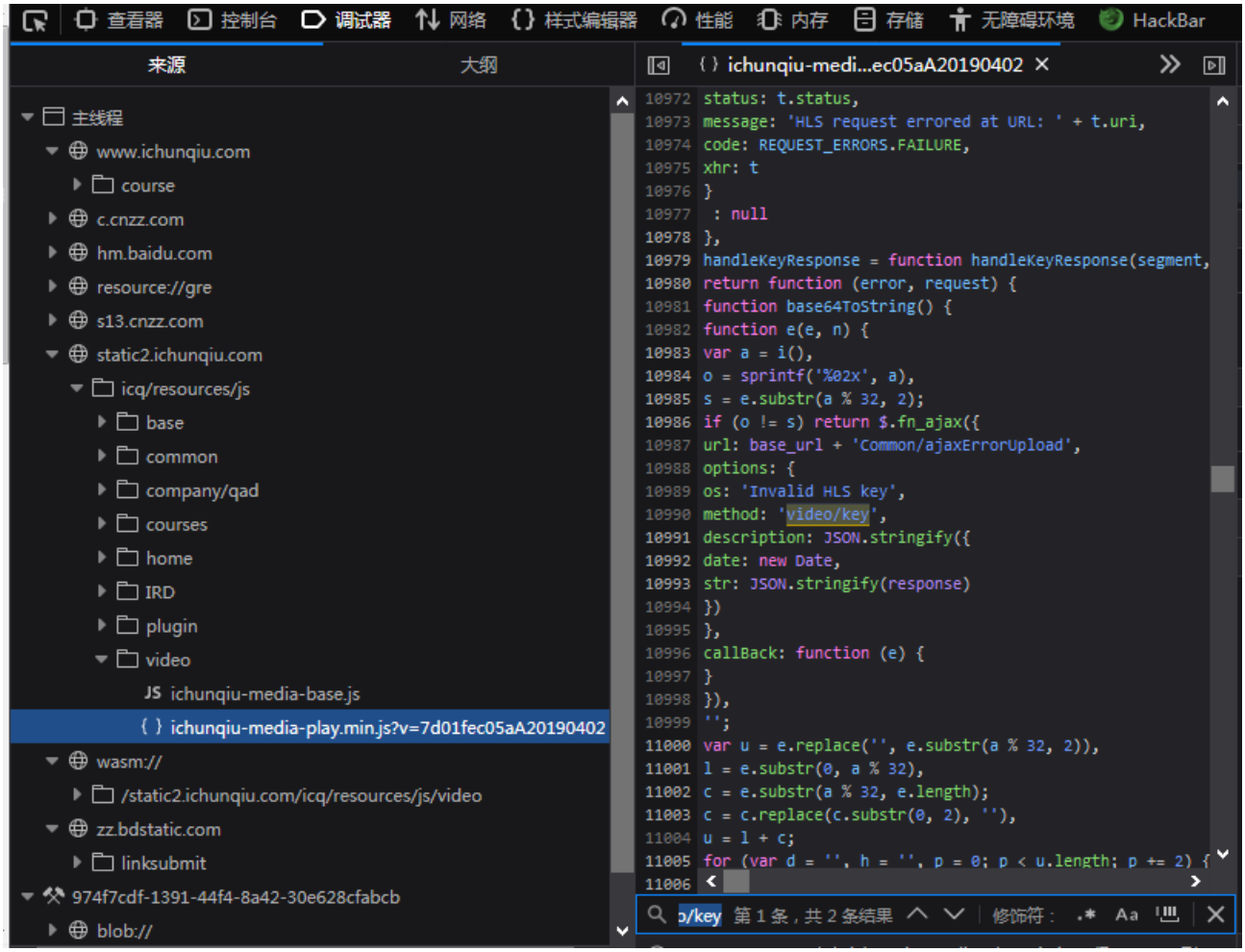
视频的内容使用AES128进行了加密，按照常理解密的密钥可通过请求
http://www.ichunqiu.com/videokey?vid=57769而得到，iv为解密的另一个参数，这里用16进制的字符串表示。
但是访问http://www.ichunqiu.com/videokey?vid=57769，发现什么响应也没有。
我们播放一个视频，看一下前端是如何获取解密密钥的，在网络请求中搜索"key"发现了一个HTTP请求很可疑



这个GET请求的url为https://www.ichunqiu.com/video/key/57769，后面这个数字与前面vid一致，应该是视频的id，响应的内容为


```
{
  "code":1,
  "message":"ok",
  "data":"UQIFAlwHWwFVXFAGVQMHBV5VAQABW1MMDAoBAQldN2VXBA==",
  "key":"U18IBQNRWVQPXAZUWFZTDFEFXVdWUAMEBwEABw5UN2VWUw==",
  "t":"12-43-09"
}
```

AES的解析密钥应该是一个16字节的字符串，在这里我猜测前端根据这些参数来生成了解密密钥，下面需要对前端加载的js进行分析，这个HTTP请求肯定是某个js发出的。
通过检索"video/key"关键字，找到了一个js文件



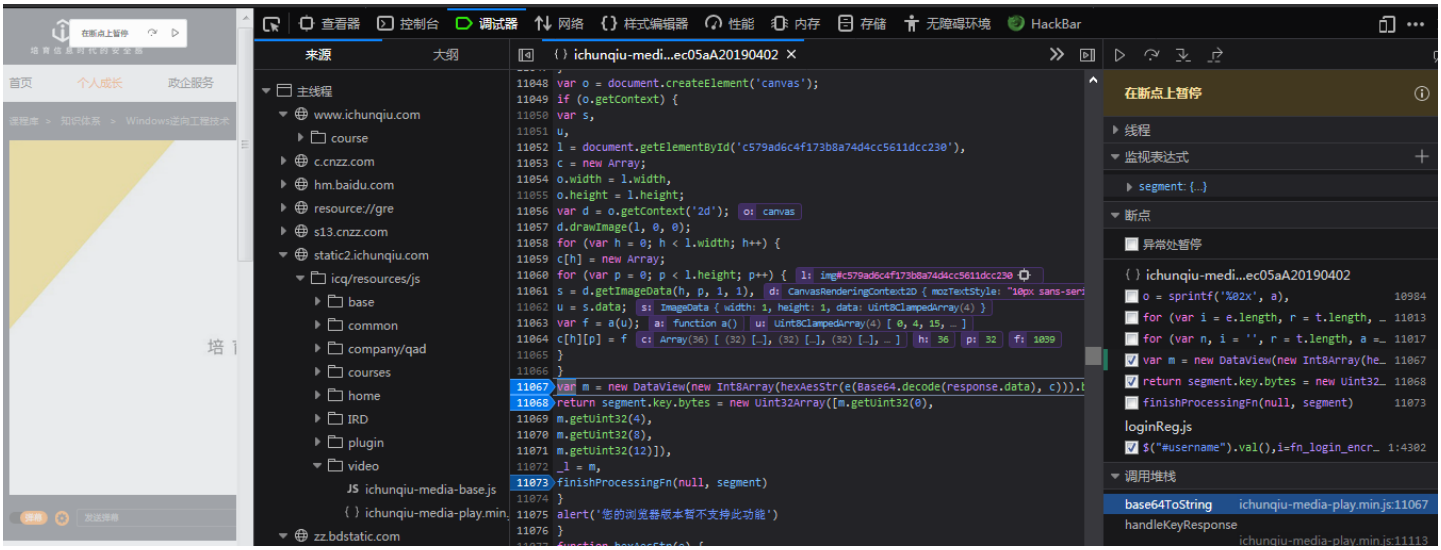
在10979行找到一个名为handleKeyResponse的函数
在11068行找到生成密钥的代码

```
11067 var m = new DataView(new Int8Array(hexAesStr(e(Base64.decode(response.data), c))).t
11068 return segment.key.bytes = new Uint32Array([m.getUint32(0),
11069 m.getUint32(4),
11070 m.getUint32(8),
11071 m.getUint32(12)]),
11072 _l = m,
11073 finishProcessingFn(null, segment)
11074 }
11075 alert('您的浏览器版本暂不支持此功能')
11076 }
```

代码为

```
var m = new DataView(new Int8Array(hexAesStr(e(Base64.decode(response.data), c))).buffer);
```

下面通过单步调试，看一下这个密钥长啥样，在11067，11068，11073行设置断点，刷新页面，页面在断点处停下了



执行一步跳到11068行
在控制点打印出局部变量看一下

```
» e(Base64.decode(response.data), c)
← "ad5192cbc9bd44a0849159f888be09d6"
»
```

这是一串16进制字符串，有可能是我们苦苦寻找的密钥，但是还需要进一步印证一下
再执行一步，代码跳到11073行
打印segment打一下（注:在mu38格式中的segment代表一个ts片断)

```
» segment
← ▼ {…}
    byterange: undefined
  ▼ key: {…}
    ▼ bytes: Uint32Array(4)
        0: 2907804363
        1: 3384624288
        2: 2224118264
        3: 2294155734
      ▶ buffer: ArrayBuffer { byteLength: 16 }
        byteLength: 16
        byteOffset: 0
        length: 4
      ▶ <prototype>: Uint32ArrayPrototype { … }
    ▼ iv: Uint32Array(4)
        0: 2578923527
        1: 3055898187
        2: 3519471670
        3: 835375451
      ▶ buffer: ArrayBuffer { byteLength: 16 }
        byteLength: 16
        byteOffset: 0
        length: 4
      ▶ <prototype>: Uint32ArrayPrototype { … }
      resolvedUri: "https://www.ichunqiu.com/video/key/57769"
    ▶ <prototype>: Object { … }
    requestId: "segment-loader-0.08570617343574816"
    resolvedUri: "https://mv.ichunqiu.com/57765/57767/57769_d/480/577690.ts"
  ▶ stats: Object { bandwidth: 828555, bytesReceived: 359904, roundTripTime: 3475 }
  ▶ <prototype>: Object { … }
```

可以看到segment中含有密钥和iv，将其转化为16进制看一下

```
In [1]: key_bytes = ['2907804363','3384624288','3384624288','2294155734']

In [2]: a = map(lambda x: hex(int(x))[2:],key_bytes)

In [3]: a
Out[3]: <map at 0x4e294a8>

In [4]: ''.join(a)
Out[4]: 'ad5192cbc9bd44a0c9bd44a088be09d6'

In [5]: iv_bytes = ['2578923527','3055898187','3519471670','835375451']

In [6]: ''.join(map(lambda x: hex(int(x))[2:],iv_bytes))
Out[6]: '99b74007b6254e4bd1c6e03631cad15b'
```

这个密钥与上面的e(Base64.decode(response.data), c)返回的结果是一致的，iv也与m3u8文本中的一致，证实了我的判断
下面来分析一下这个e函数是怎么实现的
将相关代码单独复制出来，得到下面的代码

```
function base64ToString() {
    function e(e, n) {
        var a = i(),
            o = sprintf('%02x', a),
            s = e.substr(a % 32, 2);
        if (o != s) return $.fn_ajax({
            url: base_url + 'Common/ajaxErrorUpload',
            options: {
                os: 'Invalid HLS key',
                method: 'video/key',
                description: JSON.stringify({
                    date: new Date,
                    str: JSON.stringify(response)
                })
            },
            callBack: function (e) {
            }
        }),
            '';
        var u = e.replace('', e.substr(a % 32, 2)),
            l = e.substr(0, a % 32),
            c = e.substr(a % 32, e.length);
        c = c.replace(c.substr(0, 2), ''),
            u = l + c;
        for (var d = '', h = '', p = 0; p < u.length; p += 2) {
            var f = u.substr(p, 2);
            h = '' == h ? r(o, n) : r(h, n),
                d += t(f, h)
        }
        return d
    }
    function t(e, t) {
        for (var i = e.length, r = t.length, a = '', o = 0; o < i; o++) a += n(e[o], t[o % r]);
        return a
    }
    function n(e, t) {
        for (var n, i = '', r = t.length, a = 0; a < e.length; a++) n = a % r,
            i += String.fromCharCode(e.charCodeAt(a) ^ t.charCodeAt(n));
        return i
    }
```

```
    function i() {
        var e = 180,
            t = parseInt(response.t.split('-')[0]),
            n = parseInt(response.t.split('-')[1]),
            i = parseInt(response.t.split('-')[2]),
            r = 3600 * t + 60 * n + i,
            a = r % e;
        return (r - a) / e % 128
    }
    function r(e, t) {
        for (var n, i = parseInt(e, 16), r = - 1, a = - 1, o = 0; o < 36; o++) {
            for (var s = 0; s < 32; s++) if (i == t[o][s]) {
                r = o,
                    a = s;
                break
            }
            if (r >= 0 && a >= 0) break
        }
        return n = sprintf('%02x', (7 * r + a) % 255)
    }
    function a(e) {
        var t = e,
            n = t[0],
            i = t[1],
            r = t[2];
        return parseInt(((1 << 24) + (n << 16) + (i << 8) + r).toString(16).slice(1), 16)
    }
    var o = document.createElement('canvas');
    if (o.getContext) {
        var s,
            u,
            l = document.getElementById('c579ad6c4f173b8a74d4cc5611dcc230'),
            c = new Array;
        o.width = l.width,
            o.height = l.height;
        var d = o.getContext('2d');
        d.drawImage(l, 0, 0);
        for (var h = 0; h < l.width; h++) {
            c[h] = new Array;
            for (var p = 0; p < l.height; p++) {
                s = d.getImageData(h, p, 1, 1),
                    u = s.data;
                var f = a(u);
                c[h][p] = f
            }
        }
        var m = new DataView(new Int8Array(hexAesStr(e(Base64.decode(response.data), c))).buffer);
        return segment.key.bytes = new Uint32Array([m.getUint32(0),
        m.getUint32(4),
        m.getUint32(8),
        m.getUint32(12)]),
            _l = m,
            finishProcessingFn(null, segment)
    }
    alert('您的浏览器版本暂不支持此功能')
}
```

这段代码先找到页面中一个id为c579ad6c4f173b8a74d4cc5611dcc230的图片，将其中的每个像素读取出来，经过a函数一系列的变换，赋值给一个二维数组c，再调用函数e将https://www.ichunqiu.com/video/key/57769返回的response中的data和c进行处理，得到了解密的密钥
在页面找到这个图片，发现这个图片是用base64编码的

```
<img id="c579ad6c4f173b8a74d4cc5611dcc230" src="" style="display: none;">
```

代码实现

要想使用脚本爬取所有视频，就必须实现这个生成密钥的过程。
我决定使用python来重写上面这些生成密钥的js代码，这是整个过程中最辛苦的部分，很繁锁，主要是代码中的变换我看不懂，依葫芦画瓢，运行出现了一个小bug，不过通过调试比对也轻松解决了。
get_aes_key.py

```
#-*- coding:utf-8 -*-
import base64
from PIL import Image
#全局变量
response = {}

def n(e,t):
    #print(e,type(e))
    #print(t,type(t))
    i = ''
    r = len(t)

    for a in range(len(e)):
        n = a % r
        i += chr(ord(e[a]) ^ ord(t[n]))
    return i

def t(e,t):
```

```
def t(e, t):
    i = len(e)
    r = len(t)
    a = ''
    for o in range(i):
        a += n(chr(e[o]), t[o % r])
    return a


def r(e, t):
    i = int(e,16)
    r = -1
    a = - 1

    for o in range(36):
        for s in range(32):
            if i == t[o][s]:
                r = o
                a = s
                break
        if (r >= 0 and a >= 0):
            break

    return hex((7 * r + a) % 255)[2:]


def i():
    e=180
    t,n,i = map(lambda x: int(x),response['t'].split('-'))
    r = 3600 * t + 60 * n + i
    a = r % e
    return int((r - a) / e % 128)

def e(e, n):
    a = i()
    o = hex(a)[2:].encode('utf-8')
    s = e[a%32:a%32+2]
    #print(a,o,s)

    u = e.replace(b'',s)
    l = e[:a % 32]
    c = e[a%32:a%32+len(e)]
    c = c.replace(c[:2], b'')

    u = l + c
    d = ''
    h = ''
    p = 0

    for p in range(0,len(u),2):
        f = u[p:p+2]
        h = r(o, n) if '' == h else r(h, n)
        d += t(f, h)
    return d

def a(e):
    t = e
    n = t[0]
    i = t[1]
    r = t[2]
```

```
    return int(hex((1 << 24) + (n << 16) + (i << 8) + r)[2:][1:],16)

#将base64编码的图片，保存为1.png
def save2png():
    png_bs64_string = 'iVBORw0KGgoAAAANSUhEUgAAACQAAAAgCAIAAAD1803ZAAAIjElEQVRIiQXBC/jQ470A8M/3fX8lEbqr9Pjz15nr5
FahMXK/XyK3x8gOsgdzmnOwqdlYbTw2dhzLM+cQndxqJSfMY2nluBapZJnb3CfpItnx/73v+XyIRr5Xs6N0IVeJtdwkXylmyyvEm/IIjhe7iG/oR
R9xuqa/JmtO0QyWxmu2lP+Z16WR0iJ+LZ4SP2K+WKdp5Fv5JrhfXqNdy8XSvdyt9lKvkHZRLtT0UzvV55WNYgM/U2fJF6vdlXGaHurh4hX6a88XS
8RftF/JvbQd4i35LV3viSniTnEceb3UUzNJHCY2imHyWLmIt/mC77KDNE06QXOx+Ezq4HGOlPrKjfgeIX/JC/IW8rPSn6QzWSkdLE8TK8QD0ih5q
0Z7i/yJFteo6zU7sUA7UH1Ec772Cenv2qniB+owVqozxHxBDGAP8TLfZ5P8qHaDfDl70FOaIeZwjVjFMPG1GJhYzptqN2maOFb5rnYejdhV+5F0j
7hM2lP9q3qg2IoPuV56jqKOUWeqXWwWmzXHqt2VpzleTFQ/1d6lHMMMdYQyKrFGncJ/aj+SW3G4VNS+opfaXfu59t/YnVC7lIXqcYzWrlZeE8SZ0
mz1A/U67QIe5Yeir9hN+rm8RPodl3KDGjiWa8UX8h6a63hGOlFqWcGjLJDekI/hKV6TQu4QV0n7yZs1R3OW/BR7SL2l06SPRJd8rHwzp4kDmCDPE
uOkZY1YoJ5MpzpBfUvaRixRekvnSE/yMYdptxR7s0n8XPup+Fw6Qp3HXpq3db3LE8o46US1UzpSPVRcSTf1UOkWZZ16PPskxmg+FKPFS8psdZL6e
3VnZuuarN6o9hWNdKI4UrypKeIX6t3Kb5SsjpO/zRBRxQBeFd2kZeoJdMhbiDNZIA2Svx3SbsoGuR9DlAHqIdIKZqkDpWPUPgyVbtc1XapigfZWe
X+Wa0/SPEQnuysXiINIyq7SZLU3G5SRXCIvZjoHEDSd4gmZdJkmSRulb8lzNJdIh0mf8Lb0mThEPC1Olu/g/+Tx8iua8WK5tEqaKY2T75Yf5mDpf
fkszhGTpLO5T8yV/9Zou5gidtB+IQZIh+g6QzpUeU6axlSI9/hCuZEFynr5XPVTUdXJojBVfYZX1cv4RCwWb4tO6VppkHq9/KLyonb7hk4eVrcVl
2mHM5LhymB5qLKDOopfa/vKW7FRWq8+wI/Uc5Wl0gvqntI+ykx5vTJfXam5kC919ZbeVPfS/l5MUKdKncQysVR6VTwodRNjxcfiE3GddChrNQeKn
0jD5Y2a7tIasRM96SbeF/dohkqXS/vKSR4mLed0eYpmlDxGni+/JJaII4IBzOd0eYM4Ue2n7E4PzpOOV8apU+U/KsvEEmVHVosNhPRfykHic3E/7
6qHK9eIU6VnlLP5TDpS+4Q0R9lP7lAXZ02WPpBGspt2IR+rP2au1F19UixWt+crLuG38gvSVLFK3Zk16itMYJo6Tj1WHqLOoI96IWfyqTSXu6SXW
S2uTcqZuvbVDlOHSAuZLZYxVtlTmsomzXS1uzxS2lX5lvYmthOLWCB1k06R+kuHcba2J7coywjpUSYpQ6Uz1L7qOvU3IR+gPUTaTX1IHsJf2UY7S
H5Ee7k0QDtBeln5XDpQzeoIuZ/2delXyr7ierW31EPpIT+s7ZDXajvkS5UBYrTyGR+Kd6Snk9ggvyZ2kQeqF6ifaZ+VRut6T3pfnSvvrA5mC+Vfu
UTTTxws/4T18k+l5eIi9pAW84hYp72aB7X7CMrvpF2kHeVZbMxivNrBIO0Z6vvycsaod7JB/gc3KfPVOfJAdXt5Ksexl1rFR+IN5QaxSd1Wupc+Y
h3TpJ3Ul9Tvyaeox9GhLJKnJ+UmMVbppJM3tO9yEltLY5X9xJFisRiqnCIPUx5X7lFXig71RV0/UH+m/l3qUGaznkZdzijRSsu1R6i3Sb2ZpcwM+
XzlWXE7PZReTGCk/Jx6JR9L8+ij7qmdK70i9dEu5XaxnPUqTmWW1Cme106Vr1VmK39mjrhO/VpuWKd8qQ7PYm9ppjJMbcTOmifFcawQL3Ka9gWpU
/tjebCYQZVu4yKJ8hJHq+9ym/Rb8U/KDaxWnpXule6QprFEmqXdX95RvT9zkLpBGqBulhaoI5WjxGB2Uo8RH0j/UPaSZ2kfkW5UB/GoukodKf1CX
qFMlvvqOkPaTV0szxTvqN9Rz1POkf4otlI/UR9qxHvSAPUCaa26Vh3ADOaJL5VL5VvUVWJ/5TL5LmWhtJf6ovLvTFLf1TVY/oPyurxK3Ul8LfZWB
ivHiK9EH/Ux5Q7+W/5+lkbQqDer3XhZzNQ8xpZijTpNadggtlYmin7S3tqhYrh0NpdK94np0jfqZKWLbXlQHSvtI8ZJq8XVyjL+wAfKdqHZStmsL
JWOVjaJk+QnlEXqSOku7Z/FoWKiSOpUcZ+6mSnSFdpDxWhlvdRfHc5K8aG4iO21R4mBjBXviKp9TVOUG7IyXj1dHsM8rpCOUjql98Uc7Qmig3nSf
G5WjpA3Ko+LQQp1trSZ9WK8fDkvcSq7Sr2V0zXPqP8r7af9WmyUZmq/SuwsBmsXiCvktWovaYn6K3GetI08Vx6h/UQ5Q8xXxqlvK3+SNspdDONO9
UNliKjqImW5rnmaq/gb65T/oL+4Wukjtk7SRDGRhepZyo08KQaLL7XfUQbrel2bNOOZKK8WszRna0YpSdtTfU8cSBab1OOlSfK/yP21P2Upo8VKc
YN4nhPVEaHp0H4uX6D2VKvoqb1P3lb5VCxSnpdPVhaKi9go5qoTxIF0KoO4VRypvM8KsZ6HxVJldzFbPKa00mTxkLa7vEZ9OWkvV+/XdQit+j/aL
8Sl2r9Ityg9xBTlWbGLOEzNylqpB1+p5/KGepM6iJflq8R08YDyQ+mX6oPSdpodxEnqwZptGSNt9/+pIKYt7f3+lgAAAABJRU5ErkJggg=='
    # 将 base64 字符串解码成图片字节码
    image_data = base64.b64decode(png_bs64_string)
    # 将字节码以二进制形式存入图片文件中，注意 'wb'
    with open('1.png', 'wb') as f:
        f.write(image_data)

#根据response的结果和1.png生成密钥
def get_aes_key(res):
    global response
    response = res
    img=Image.open("1.png")
    img_array=img.load()
    width,height = img.size
    data = []
    for i in range(width):
        i_v = []
        for j in range(height):
            i_v.append(a(img_array[i,j]))
        data.append(i_v)

    return (e(base64.b64decode(response['data']), data))
```

获取了密钥，下载视频就很简单了，其它部分的代码如下。

爬虫的代码如下

ichunqiu_spider.py

```
#-*- coding:utf-8 -*-
```

```python
#ichunqiu_spider.py
#使用方法 python ichunqiu_spider.py [网课的地址]
#如python ichunqiu_spider.py https://www.ichunqiu.com/course/57769
import requests
from requests.packages import urllib3
from Crypto.Cipher import AES
import re
from bs4 import BeautifulSoup
from get_aes_key import get_aes_key
import os,sys
urllib3.disable_warnings()

#全局变量，requests请求所使用headers字段
headers = {
    'User-Agent':'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0'
}

#全局变量，保存requests会话
s = requests.Session()

#用来扩展AES密钥，如果value的长度不足16个字节，在其后面填充0
def fill_character(value):
    if len(value) < 16:
        value = value.ljust(16, '\000')
    elif len(value) > 16:
        value = value[:16]
    return value

#获取视频加密所使用的AES密钥
def get_key(video_id):
    key_url= 'https://www.ichunqiu.com/video/key/%s' % video_id
    res1 = s.get(key_url,headers=headers,verify=False)
    if res1.status_code == 200:
        response = res1.json()
        #根据response中的值来生成密钥
        return get_aes_key(response)
    return ''

#根据iv和key对ts视频内容进行解密
def decrypt_single_ts(ts,iv_str,key_str):
    #将iv由十六进制字符串转化为byte
    iv = bytes.fromhex(iv_str)
    #将密钥由十六进制字符串转化为byte
    key = bytes.fromhex(key_str)
    #计算需要填充的字节长度
    pad_len = AES.block_size - len(ts) % AES.block_size
    #若ts长度不是的AES要求的分组长度的整数倍，对其填充0
    if pad_len != AES.block_size:
        ts = ts[:-pad_len] + bytes([0] * pad_len)
    #解密操作
    cipher = AES.new(key, AES.MODE_CBC, iv=iv)
    out_data = cipher.decrypt(ts)
    #从解密结果中去掉填充部分，得到ts的解密内容
    if pad_len != AES.block_size:
        out_data = out_data[:-pad_len]
    return out_data

#根据m3u8视频流的url，爬取视频内容
#m3u8_url为m3u8的地址，title为视频保存的文件名
def handle_m3u8_data(m3u8_url,title):
```

```
    res = s.get(m3u8_url,headers=headers,verify=False)
    if res.status_code == 200:
        #这是一个文本文件
        data =  res.text.strip()
        #print data
        #使用正则表达式，提取出加密方法，视频id，和iv
        aes_method,video_id,iv_str = re.findall(r'#EXT-X-KEY:METHOD=(.*?),URI="http:\/\/www.ichunqiu.com\/videok
ey\?vid=(\d+)",IV=0x(.*?)\n',data)[0]
        #使用正则表达式提取来ts的uri
        ts_uri_list =  re.findall(r'(\d+.ts)\n',data)
        #根据video_id获取解密密钥
        key_str = get_key(video_id)
        print(key_str)
        print(iv_str)
        content=b''
        #下载所有的ts文件
        for ts_url in ts_uri_list:
            #构造完成的url
            url_base = m3u8_url[:m3u8_url.rfind('/')+1]
            res1 = s.get(url_base+ts_url,headers=headers,verify=False)
            if res1.status_code == 200:
                #对ts的内容进行解密，并依次拼接成一个完整的文件
                content += decrypt_single_ts(res1.content,iv_str,key_str)
        #保存输出文件
        open('%s' % title,'wb').write(content)

#爬虫主函数
def spider():
    global s
    #登陆url
    url = 'https://user.ichunqiu.com/login/normal'
    #登陆信息
    data = {
        'redirect_url':"https://www.ichunqiu.com/",
        'appid':'5af018bda55004e1',
        'account':'xxxxxxxxxxxxxxxxxxxxxxxxxxx',
        'password':'xxxxxxxxxxxxxxxxxxxxxxxxxxx',
        'captcha':'',
        'mt':'1577261775197',
        'rs':'e8c48853ab79882bc54ef7f6b5452c98'
    }
    res = s.post(url=url,data=data,headers=headers,verify=False)
    if res.status_code == 200:
        #若登陆成功
        if res.json()['code'] == 0:
            #m3u8_url = 'https://mv.ichunqiu.com/144/279/147_d/147.m3u8'
            #m3u8_url = 'https://mv.ichunqiu.com//57765/57767/57769_d/320/57769.m3u8'
            #course_url = 'https://www.ichunqiu.com/course/57765'
            #课程主页的url
            course_url = sys.argv[1]
            res1 = s.get(course_url,headers=headers)
            if res1.status_code == 200:
                #使用bs4库来解析html文档
                soup = BeautifulSoup(res1.text,'lxml')
                #课程的名字
                course_name = soup.title.text.strip().split('_')[0]
                print(course_name)
                for video_catalog_list in soup.find('div',class_='video_catalog').find_all('div',class_= 'video_
catalog_list'):
```

```
                    #课程中章的名字
                    zhang_title = video_catalog_list.find('div',class_='v_catalog_zhang listName')['title'].stri
p()
                    print(zhang_title)
                    #构造存储路径
                    path = 'data\\%s\\%s'%(course_name,zhang_title)
                    #若该目录不存在，创建该目录
                    if not os.path.isdir(path):
                        #os.makedirs会递归创建目录，os.mkdir只会在当前目录下创建目录
                        os.makedirs(path)
                    #找到当前章下面的课时
                    for a in video_catalog_list.find_all('a',class_='_courseList'):
                        #课时的title
                        keshi_title = a['title']
                        #课时页面的url
                        keshi_url = a['href']
                        print(keshi_title)
                        #请求课时页面
                        res2 = s.get(keshi_url,headers=headers,verify=False)
                        if res2.status_code == 200:
                            #利用正则表达式找到m3u8地址
                            m3u8_url1 = re.findall(r'data-video-url="(https.*?m3u8)',res2.text)[0]
                            index = m3u8_url1.rfind('/')
                            #一个视频有多个清晰度选项，选择720P，构造url
                            m3u8_url = m3u8_url1[:index+1]+'720/'+m3u8_url1[index+1:]
                            print(m3u8_url)
                            #保存的视频文件名
                            filename = '%s\\%s.mp4' % (path,keshi_title)
                            #若该文件不存在
                            if not os.path.isfile(filename):
                                try:
                                    #下载该视频
                                    handle_m3u8_data(m3u8_url,filename)
                                except Exception as e:
                                    #出错的话不处理
                                    print(e)

                            print('下载完毕')


spider()
```

参考资料

- m3u8解密
  https://blog.csdn.net/weixin_42572656/article/details/96981292
  https://www.zuowting.top/2018/Python%E8%A7%A3%E5%AF%86m3u8%E8%A7%86%E9%A2%91.html
- AES加解密
  https://blog.csdn.net/Vieri_32/article/details/48345023
- 火狐如何调试js
  https://developer.mozilla.org/zh-CN/docs/Tools/Debugger
- 如何调试python
  https://www.cnblogs.com/jing1617/p/9396617.html