

爬虫反爬 - JS 逆向 02 - G某戏 加密 解密 - RSA加密

[返回首页](#)

什么是逆向：

逆向工程（又称逆向技术），是一种产品设计技术再现过程，即对一项目标产品进行逆向分析及研究，从而演绎并得出该产品的处理流程、组织结构、功能特性及技术规格等设计要素，以制作出功能相近，但又不完全一样的产品。逆向工程源于商业及军事领域中的硬件分析。其主要目的是在不能轻易获得必要的生产信息的情况下，直接从成品分析，推导出产品的设计原理。

逆向工程可能会被误认为是对知识产权的严重侵害，但是在实际应用上，反而可能会保护知识产权所有者。例如在集成电路领域，如果怀疑某公司侵犯知识产权，可以用逆向工程技术来寻找证据。

爬虫反爬 - JS 逆向 02 - G某戏 加密 解密 - RSA加密 定期练习 定期提高

网站链接：<http://www.opcd.com.cn/login.html?backUrl=/index>



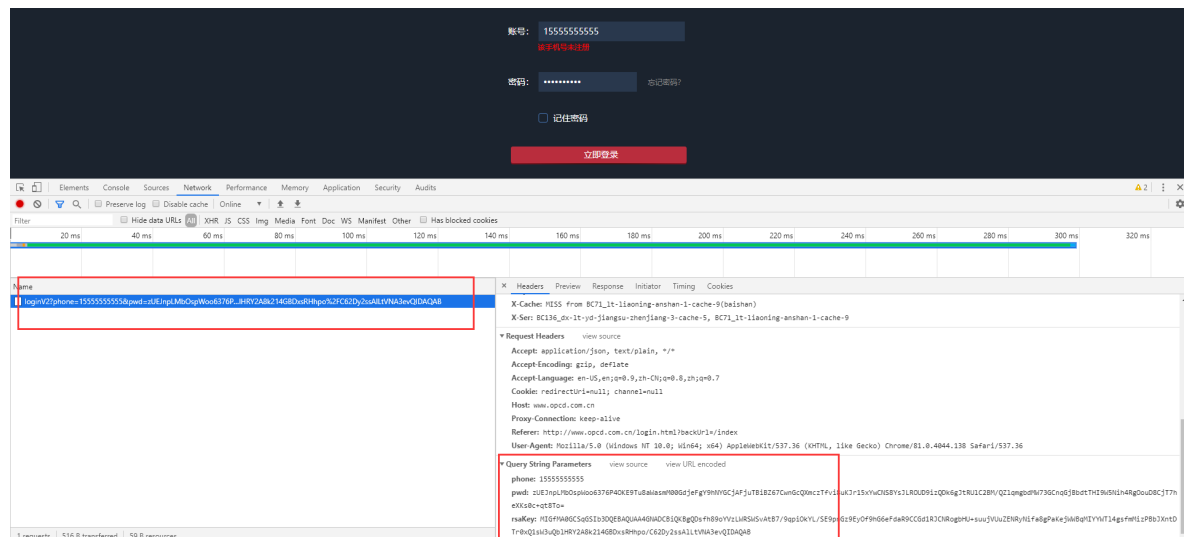
老规矩 第一步 直接祭出我们的大招：

肉眼观察法 一个账号一个密码

话不多说 直接抓包:



老套路了 这里就不多解释了 直接抓包看看



这个网站很简单 直接一个包就抓到了

他这几个参数当中 观察一下 有一个很细节的东西 `rsaKey`

phone: 15555555555

pwd:

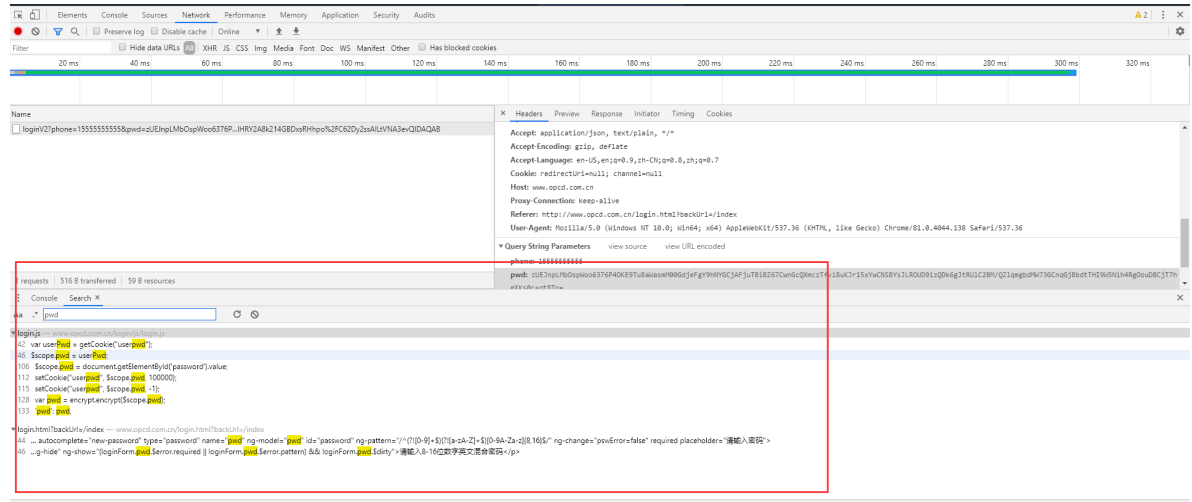
zUEJnpLMBospwoo6376P4OKE9Tu8awasmM00GdjefgY9hNYGCjAFjuTBiBZ67CwnGcQXmczTfvi8uKJr
15xYWCNS8YSJLROUD9iziQdK6gJtRU1C2BM/QZ1qmgbdMW73GcnqGjBbdtTHI9W5Nih4RgOouD8CjT7he
Xks0c+qt8To=

rsaKey:

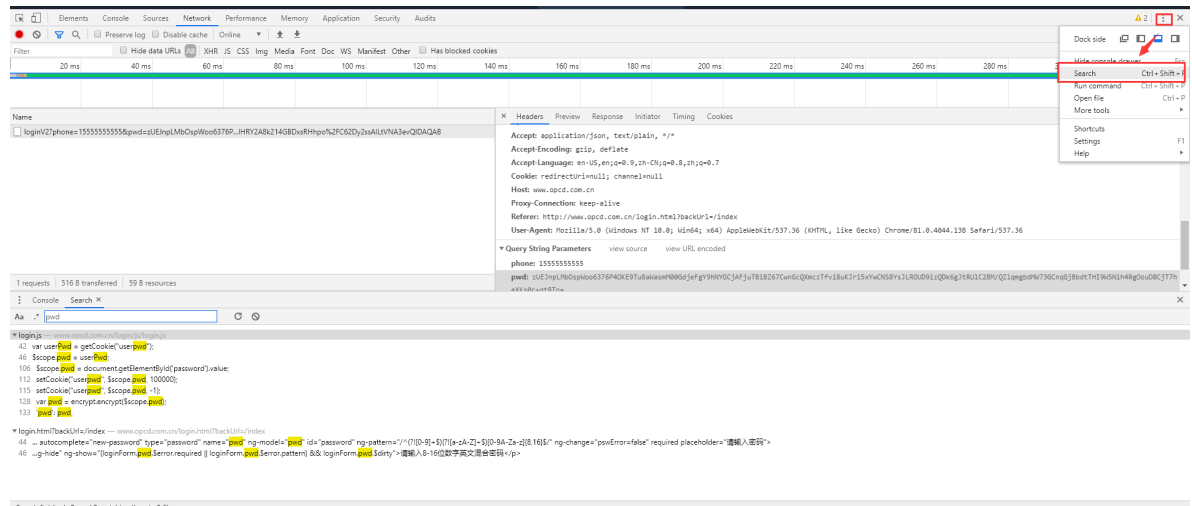
MIGfMA0GCsqGSib3DQEBAQUAA4GNADCBiQKBgQDs fh89oYVzLWRSWSvAtB7/9qp iokYL/SE9pnGz9Eyo
f9hg6eFda9CCGd1RJCNRogbHU+suujVUuZENRYni fa8gPaKejWWBqMIYYWT14gsfmMizPBbJXntDTR0
xQ1sw3uQb1HRY2A8k214GBDxsRHhp/C62Dy2ssa1LtVNA3evQIDAQAB

那么就可以初步断定 这个加密是用大名鼎鼎的 RSA 里边的 pwd 密文 应该就是用 RSA 进行加密了

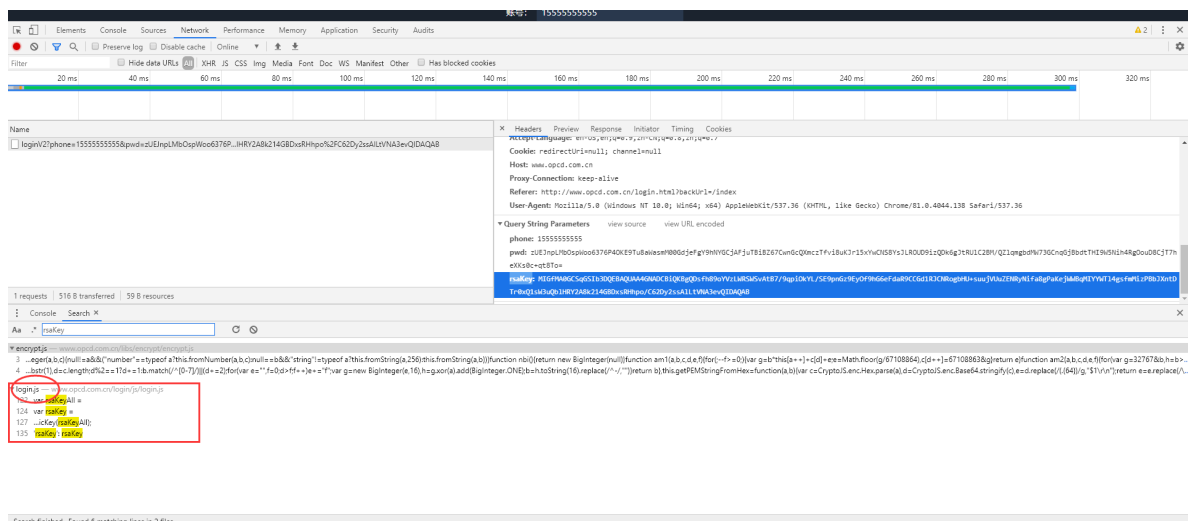
我们直接打开全局搜索框 搜索关键参数



哦对 还记得怎么打开这个全局搜索框吧?? 这里就多嘴一句 从这里打开

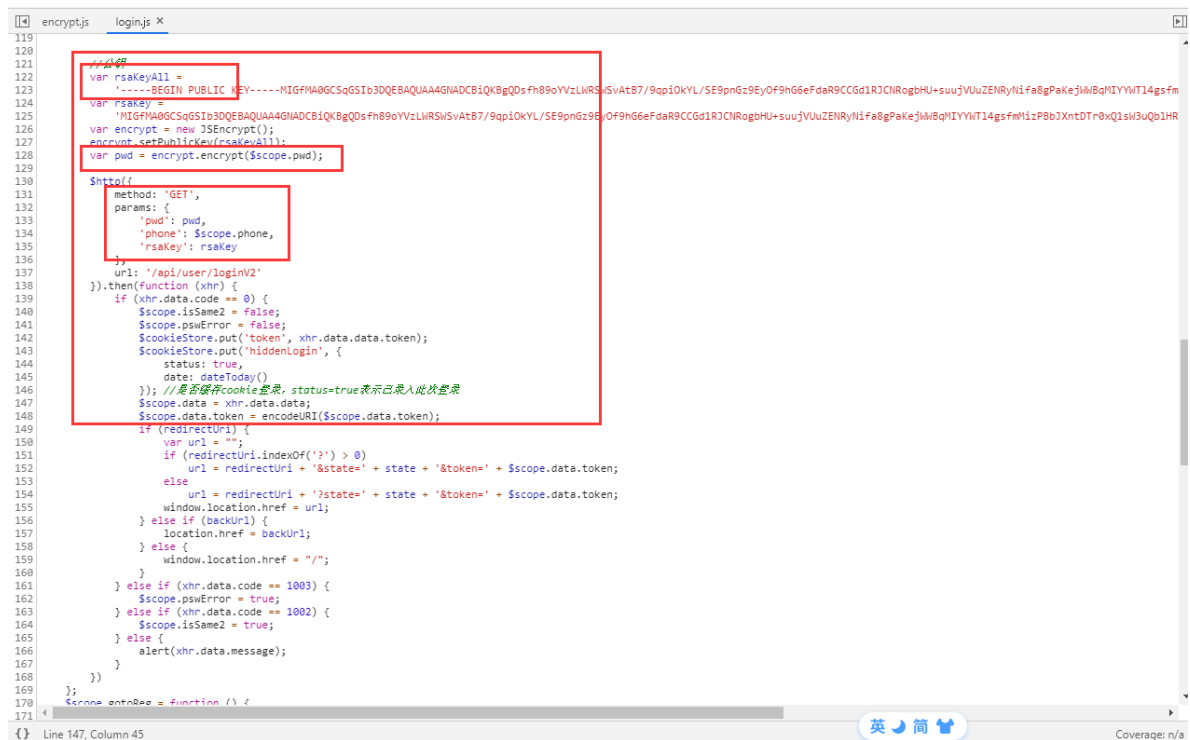


看了这里的参数 感觉很多… 不想找 我们换一个参数试试 换 rsaKey 搜索一下看看



这个看起来不错 点进去看看 这个要比刚才的 `pwd` 关键字要好的多

点进去之后 确实直接到了加密的地方 而且确认是 `RSA` 加密



第一个框里边有 `BEGIN PUBLIC KEY` 这种关键字 那就可以断定是 `RSA` 加密了

第二个框是加密的方法（可以打上断点看看结果是不是我们想要的）

第三个框就是HTTP请求了 发送包

这里先介绍一下RSA加密 磨刀不误砍柴工(网上有人说的挺好 我就拿来了)

RSA加密是一种非对称加密。可以在不直接传递密钥的情况下，完成解密。这能够确保信息的安全性，避免了直接传递密钥所造成的被破解的风险。是由一对密钥来进行加解密的过程，分别称为公钥和私钥。两者之间有数学相关，该加密算法的原理就是对一极大整数做因数分解的困难性来保证安全性。通常个人保存私钥，公钥是公开的（可能同时多人持有）。

RSA的加密过程如下：

- 1.A生成一对密钥（公钥和私钥），私钥不公开，A自己保留。公钥为公开的，任何人可以获取。
- 2.A传递自己的公钥给B，B用A的公钥对消息进行加密。
- 3.A接收到B加密的消息，利用A自己的私钥对消息进行解密。

在这个过程中，只有2次传递过程，第一次是A传递公钥给B，第二次是B传递加密消息给A，即使都被敌方截获，也没有危险性，因为只有A的私钥才能对消息进行解密，防止了消息内容的泄露。

PS:RSA加密对明文的长度有所限制，规定 加密的明文最大长度=密钥长度-11（单位是字节，即byte），所以在加密和解密的过程中需要分块进行。而密钥默认是1024位，即1024位/8位-11=128-11=117字节。所以默认加密前的明文最大长度117字节，解密密文最大长度为128字。那么为啥两者相差11字节呢？是因为RSA加密使用到了填充模式（padding），即内容不足117字节时会自动填满，用到填充模式自然会占用一定的字节，而且这部分字节也是参与加密的。

下面我们可以到一个网站来玩一玩RSA：

BEJSON 下面的链接可以尝试用RSA进行加密 体验一下加密流程

<https://www.bejson.com/enc/rsa/>

BEJSON 工具 编码/加密 格式化 网络 前端 后端 转换 其他 文档 平台工具 更多 功能搜索

Base64加密、解密 MD5加密 MD5解密 DES,AES等对称加密解密 RSA,RSA2公钥私钥加密解密 SHA加密 Escape编码 URL编码 更多

什么是JSON JSON的用法 JSON在线编辑 七牛云(99元/年) 香港/美国服务器(37元/月) JQuery常用代码 JQuery文档 阿*里*云3折

RSA,RSA2公钥私钥加密解密

RSA公钥生成 根据公钥加密文本 根据私钥解密文本

密钥长度 2048 bit 密钥格式 PKCS#1 私钥密码 可以为空 生成公私密钥

RSA加密公钥 复制公钥 RSA加密私钥 复制私钥

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwSPuFIUXSL16Q4Npa2X
YwkBuoyONOhbA1D+J0sk7Qckf0hxh8I2XyCqWHD00sOEPcIN6fz41kdvaFVFKTC
COUB4M2tX1KJedVfDTYT+znLEcNC5uOtWqJgdN9svm7m79oBk0M8UARmga/hrY/
ikbG6J2DWggTSY+kxPLtOs8TRpgJ18ytn3gkuDEf35Ah2ZKmoCmiz1PkKF9GrSnI
HdWvqtL2ANbqJLIAVpB8xE1swsHE1kPohGymNBdVceDEaqTUGDTtB15xxEnpskxK
m+LrF0CsJgA8WL8U911SMThDozRhuJtF2dA8Kwm7Z8nbvLoBuaMh63rbMHqaxbuP
EQIDAQAB
-----END PUBLIC KEY-----

-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC9ZI+4UhRdlvXp
Dg2lrZdJCQ6jKg06FsDUP4nSyTtByR/SHGHwjZHKrAd3Q6w4Q8KU3p/PjWR29o
VUVUpNw15QHgzaf1Uol4NV8O1hP7OelsRw0Lm461aomB032xWbubv2gGTQzxQBG
a
Br+Gtj+KRsbonyNaCC1Jj6TGku06zxNGmAnXzK2feCS4MR/fkCHZkqagKaLPu+Qo
X0aueyUd1a+q0vYA1uoksgBWkHzeSWzCyETWQ86EbKY0F1Vx4MRqpNSANOOGXnf
c
SemyTEqb4usXQKwmADxYvxTZWVlxOEOJNGG4m0XZ0DwrCbtynydu8ugG5oyHretsw
epRFu48RAgMBAAECggEAUHTK/TeZQQu2DgCTWZvKIR9VZauVYyUBW+12II6sVsE
bTiyUMT3lifoaaP9E221wi0FoW57O4RvVoZK8a9abiiiOF4H1ePZVIDP1hvatmMb

您最近使用了: RSA,RSA2公钥私钥加密解密

JSON工具 编码/加密 格式化 网络 前端 后端 转换 其他 文档 平台工具 更多

- 免费JSON接口
- BLogs
- 新浪天气API接口测试用例
- sitemap提交入口大全
- 留言
- 备案查询
- 更新日志
- 常用软件下载
- 广告招租

点击蓝色的按钮 即可自动生成公钥和私钥：

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwSPuFIUXSL16Q4Npa2X
YwkBuoyONOhbA1D+J0sk7Qckf0hxh8I2XyCqWHD00sOEPcIN6fz41kdvaFVFKTC
COUB4M2tX1KJedVfDTYT+znLEcNC5uOtWqJgdN9svm7m79oBk0M8UARmga/hrY/
ikbG6J2DWggTSY+kxPLtOs8TRpgJ18ytn3gkuDEf35Ah2ZKmoCmiz1PkKF9GrSnI
HdWvqtL2ANbqJLIAVpB8xE1swsHE1kPohGymNBdVceDEaqTUGDTtB15xxEnpskxK
m+LrF0CsJgA8WL8U911SMThDozRhuJtF2dA8Kwm7Z8nbvLoBuaMh63rbMHqaxbuP
EQIDAQAB
-----END PUBLIC KEY-----
```

-----BEGIN PRIVATE KEY-----

```
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC9ZI+4UhrDvXp
Dg2lRzdjCQG6jkg06FsDUP4nSyTtByR/SHGHwjZfIKrAd3Q6w4Q8KU3p/PjWR29o
VUVUPnWi5QHgza1fu0l4NV801hP70eIsRw0Lm461aomB032xwbubv2gGTQzxQBGa
Br+Gtj+KRSbonYNaCC1j6TGku06zxNGmAnXzK2feCS4MR/fkCHZkqagKaLPU+Qo
X0auyeUd1a+q0vYA1uoksgBwkhZESWzCyETWQ86EbKY0F1Vx4MRqpNSAN0GXnFc
SemyTEqb4usXQKwMADXYvXT2WVIX0EOjNGG4m0XZ0DwrCbtndydu8ug5oyHretsw
eprFu48RagMBAAECggEAUHTk/TeZQQu2DgCTWZvKIR9VZauVVyyUBw+72I76svSe
bTIYUMT3lifpggP9E221w10Fpw57Q4RrvpZK8q9qbjjjOF4H1ePZVIDP1hyptmMb
qxwv/Atkb7tHmSiAUBQ/t+VzSqH26pH+KEHPowNqymxQ2cb4YwnwOzzTkiwerYlJ
oEtqtnH9NPLFCgFuSLEwRLyv4wECehd0gZpO/dk1u6d9Mx17TXMMWEBTTIUANzki
+K0N3vxHQBPzCkIhIaHoQY5MvywgTd8iD2N557Q0xclyz3UjMx8tDnkgFCo6rX1
HFLHiPVxZLA1A2iWEFpofDFGisiAgRDYeTHiXoos5QKBgQDc3/D5UuHFLimUhTlC
PczQra46dN5H2tz1SD6LH/NpTYftqu7wiY43+HJBEwvONJlyL+yPSKoJOoKnz+dR
oR/htJrQd5Si3LesJ0b5nJz6EcbZqtfuAkhe14M1A9FQZTECi80vkrOCEZxiobqT
15ckiFX9Pcy9A7yg8vXnFW3YtwKBgQDbgVrixTiz7kMDCKhJ95ODI15tHB+ZyLAO
ApMFnsXqLCue7NK0dAzjK66UV7LFBgp13Q2UBww86cFzikpwOU+p+qkr1/OhGp1k
0VfVbAd0WJEC/jgLSiKInHBC3qZpubjb1SwJPd147twvjGpN9y3MVoTB6e3Zx6D
ahCwIcy+dwKBgQCuvQWEkIjTBPwhNPnThTtFLiZm3S6Wys74ZAcihZkgJ5xJMRHI
iphkPp2KL5DRW6T06WE7VT78/fqu/odIVjb4IhknDJu3h2eA7ErBIk9av00VXu1
pung0T0dJV3+gL9hTI23XJ4fMAZs8SY4yvTQOut65u+prOvwosxveiHHzwKBgBEE
uR/DPJ8nonDHRgzeBfC/2D1LvePdXFDpwc5KPbthKyY3pCpbaPhf1ZwopHSH6hvV
pIzacwdx9A6aJSNENDnwdZYqJLhT0Jqxy8XkpztM9osx5nRxnfxLCyawGGTH8jYQ
8+YPhoHKuoRu141EEP+duudwy377nrD/kywkqTQHAoGASUQRJvhIdB2Ar+bhu+D1
6wd8WPLZebiIapN3WR9Gjkf02N1JV2E21mjuan5CmbkjT8Z2FQZUujvFiS3trqat
yCexIgcIC/qAn/54nAry0JLFDNu7+ceijCqZGRpsX+O+jnIMG4NU1zjikiy2+i5n
Jt9AXQ5B77ymJ75xhMvBiPI=
```

-----END PRIVATE KEY-----

什么是JSON JSON的用法 JSON在线编辑 七牛云 腾讯云(99元/年) 香港/美国服务器(37元/月) JQuery常用代码 JQuery文档 阿*里*云3折

RSA,RSA2公钥私钥加密解密

RSA公钥生成 根据公钥加密文本 根据私钥解密文本

请输入公钥

```
2 MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuWSPuFIUXSL16Q4Npa2X
3 YwkBuoyoNOhba1D+J0sk7Qckf0hxb8I2XyCqwHd0OsOEPcIN6fz41kdvaFVVKTC
4 COUB4MztX1KJeDVfDTYT+znLEcNC5uOtWqJgdN9sVm7m79oBk0M8UARmga/hrY/
5 ikbG6J2DWggtSY+koPLtOs8TRpgJ18ytn3gkuDEf35Ah2ZKmoCmiz1PkKF9Grsl
6 HdWvqtL2ANbqJLIAVpB8xElswshE1kPOhGymNBdVceDEaqTUgDTtBI5xXEnpskK
7 m+LrF0CsJgA8WL8U9IISMThDozRhuTf2dA8Kwm7Z8nbvLoBuaMh63rbMHqaxbuP
8 EQIDAQAB
9 -----END PUBLIC KEY-----
10
```

请输入要加密的字符串

```
1 http://www.wdkang.top
```

RSA2 执行 清空 下载加密/解密代码 复制加密/解密代码

```
1 Bqdxesb2rSW26eVqMty2LI/bNjvW6e1Bq4WdwWa2QECoHEP6XSDosPhtKNE8x6sO9Lt7nP8U47txGD2ZPVV7n2xBBXRSFbSbGRicMssHVVA54r/f2s1jt8ufvauxWXGgTI
/rDnmthKciNY2Vi/qRYhMAfuZHS9rjPySDk/GTYMMmQOCy4L3IOE8oSa/bbDCrEQbyA+3fKhAoiMa31IsUWeyxmORX9ez1sBl3r0Owtkvs6GhHikl8MOA0yeaSIliID8auQJOI
U+GqEQ6uqAkwZowFHZAV/is7LWcfArxCVzeLxEuKYUcdFsEBr+eVA20JL6nSn56toM2LNH4nly+Q==
```

您最近使用了: RSA,RSA2公钥私钥加密解密

填写好公钥（私钥要保存好的哦 不能公开的）

然后填写好我们要加密的内容

蓝色小按钮执行后 返回就是我们要的密文

```
Bqdxesb2rSw26evQmTy2Ll/bNjvW6e1Bq4WdvWa2QECCoHEP6XSDosPhtKNE8x6sO9Lt7nP8IJ47txGD
2ZPVV7n2xBBXRSFbSbGRicMssHVVA54r/f2s1ijt8ufvauXWGgTI/rDnmthKciNY2Vi/qRYhMAfuZHS
9rjPySDkGTYMMmQOCy4L3lOE8oSa/bbDCrEQbyA+3fKhAoiMa31lSUweyXmORX9ez1sBl3r0Owtkvs6
GhHiki8MOA0yeaSlIId8auQjOiU+GqEQ6u/akZowFHZAV/is7LWcfArxCVzeLxZEuKYUcdFsEBr+eVA2
OJL6nSn56toM2LNH4nIy+Q==
```

接着我们用私钥把他解密

RSA,RSA2公钥私钥加密解密

RSA公钥生成 根据公钥加密文本 根据私钥解密文本

请输入私钥

```
21 pUng0T0dJV3+gL9hTi23X74fmAZs8SY4yvTQOUt65u+pROvwosxveiHHzwKBgBEE
22 uR/DPJ8nonDHRgzeBfC/2D1LVePdXFDpwC5KPbtHKyY3pCpbaPhf1ZwopHSH6hvV
23 plzacwdx9A6aJSNENDnwdZYqJLhT0JqxY8XkpztM9osx5nRxnfxLcyawGGTH8jYQ
24 9+YPhoHkueRu14IEEp+dUudwy377nrD/kyWkqTQHAAoGASUQRUVhldB2Ar+bhu+DI
25 6Wd8WPLzEbilapN3wR9GjktO2N1JV2E21mJuan5CMbkjT8Z2FQZUJvFIS3trqat
26 yCexlgclC/QaN/54nAry0JLFDNu7+cejCqZGRpSX+O+jnlMG4NU1zjiKiy2+i5n
27 Jt9AXQ5877ymJ75xhMvBIPI=
28 -----END PRIVATE KEY-----
29
```

请输入要解密的签名

```
1 Bqdxesb2rSw26evQmTy2Ll/bNjvW6e1Bq4WdvWa2QECCoHEP6XSDosPhtKNE8x6sO9Lt7nP8IJ47txGD2ZPVV7n2xBBXRSFbSbGRicMssHVVA54r/f2s1ijt8ufvauXWGgTI
/rDnmthKciNY2Vi/qRYhMAfuZHS9rjPySDkGTYMMmQOCy4L3lOE8oSa/bbDCrEQbyA+3fKhAoiMa31lSUweyXmORX9ez1sBl3r0Owtkvs6GhHiki8MOA0yeaSlIId8auQjOi
U+GqEQ6u/akZowFHZAV/is7LWcfArxCVzeLxZEuKYUcdFsEBr+eVA2OJL6nSn56toM2LNH4nly+Q==
```

RSA2 执行 清空 下载加密/解密代码 复制加密/解密代码

```
1 http://www.wdkang.top
```

您最近使用了: RSA,RSA2公钥私钥加密解密

点击蓝色的小按钮 就可以看到我们加密前的内容啦!

这就是RSA加密

那根据上边的操作 我们应该需要找的是:

- 1.需要加密的内容: 我们的密码 a123456789
- 2.PUBLIC KEY 公钥 上边已经看到了

```

119
120
121
122 var rsaKeyAll =
123 "-----BEGIN PUBLIC KEY-----MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfm
124 var rsaKey =
125 "MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfmMizPbDjXntDT0xQ1sh3uQb1HR
126 var encrypt = new JSEncrypt();
127 encrypt.setPublicKey(rsaKeyAll);
128 var pwd = encrypt.encrypt($scope.pwd);
129
130
131 $http({
132   method: 'GET',
133   params: {
134     'pwd': pwd,
135     'phone': $scope.phone,
136     'rsaKey': rsaKey
137   },
138   url: '/api/user/loginV2'
139 }).then(function (xhr) {
140   if (xhr.data.code == 0) {
141     $scope.isSame2 = false;
142     $scope.pswError = false;
143     $cookieStore.put('token', xhr.data.data.token);
144     $cookieStore.put('hiddenLogin', {
145       status: true,
146       date: dateToday()
147     }); // 是否保存cookie登录, status=true表示已登录人此次登录
148     $scope.data = xhr.data.data;
149     $scope.data.token = encodeURI($scope.data.token);
150     if (redirectUrl) {
151       var url = "";
152       if (redirectUrl.indexOf('?') > 0)
153         url = redirectUrl + '&state=' + state + '&token=' + $scope.data.token;
154       else
155         url = redirectUrl + '?state=' + state + '&token=' + $scope.data.token;
156       window.location.href = url;
157     } else if (backUrl) {
158       location.href = backUrl;
159     } else {
160       window.location.href = "/";
161     }
162   } else if (xhr.data.code == 1003) {
163     $scope.pswError = true;
164   } else if (xhr.data.code == 1002) {
165     $scope.isSame2 = true;
166   } else {
167     alert(xhr.data.message);
168   }
169 });
170 $scope.enter = function () {

```

第一个框就是 PUBLIC KEY

第二个框 \$scope.pwd 就是我们的密码

这里虽然我们已经确定他是 RSA 加密了 这里可以直接用RSA的库来模拟RSA加密

但是更多时候 我们很可能是看不出来加密方法的 所以我们最直接的方法就是把他整个加密方法都扣出来

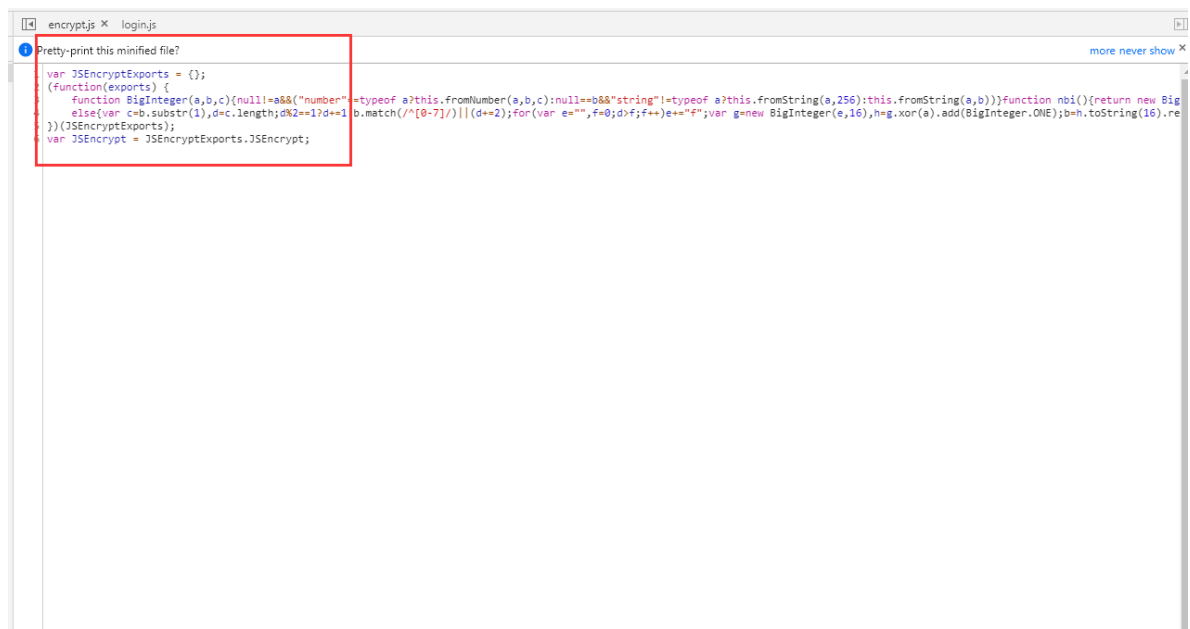
上图的第126行有一个 var encrypt = new JSEncrypt(); 我们可以在下边打一个断点 接着进入 new JSEncrypt() 这个对象中 将里边的内容给扣出来 最后结合它的代码 将加密的结果 运行出来

```

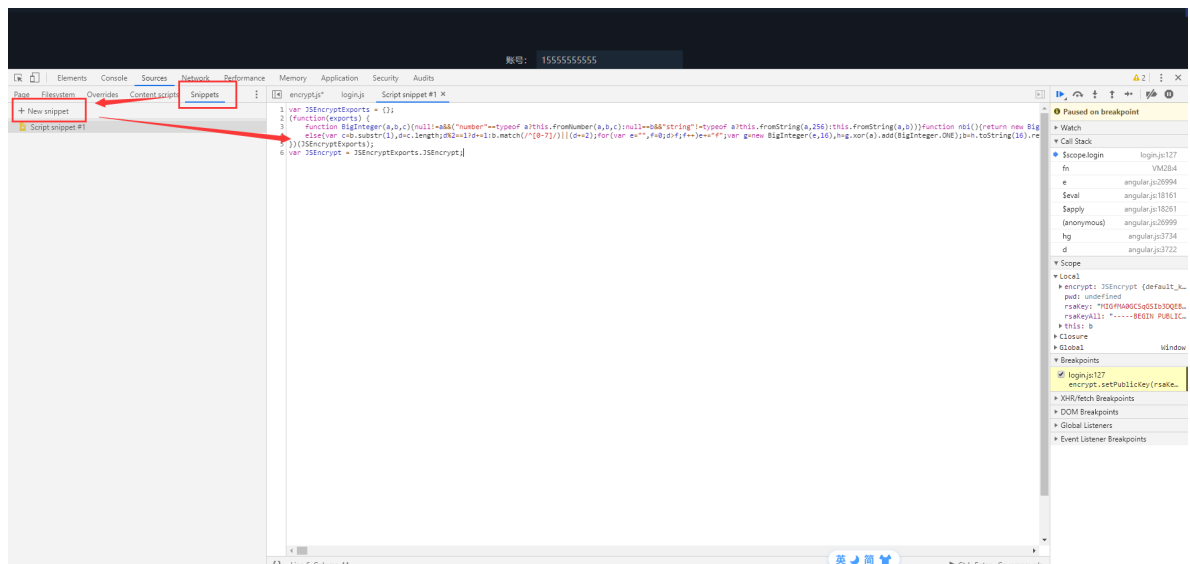
112 setCookie("userpwd", $scope.pwd, 100000);
113 } else {
114   setCookie("username", $scope.phone, -1);
115   setCookie("userpwd", $scope.pwd, -1);
116 }
117
118
119
120
121 //公钥
122 var rsaKeyAll = "-----BEGIN PUBLIC KEY-----MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfm
123 "-----BEGIN PUBLIC KEY-----MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfm
124 var rsaKey = "-----BEGIN PUBLIC KEY-----MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfm
125 "MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKgQDsfn89oYVzLWRSW5vAtB7/9api0kYL/SE9pnGz9EyoF9HG6eFdaR9CCGd1R3CNRogbHU+suuJVUuZENRyNif8gPaKejMhBqMIYVNT14gsfmMizPbDjXntDT0xQ1sh3uQb1HR
126 var encrypt = new JSEncrypt();
127 encrypt.setPublicKey(rsaKeyAll);
128 var pwd = encrypt.encrypt($scope.pwd);
129
130
131 $http({
132   method: 'GET',
133   params: {
134     'pwd': pwd,
135     'phone': $scope.phone,
136     'rsaKey': rsaKey
137   },
138   url: '/api/user/loginV2'
139 }).then(function (xhr) {
140   if (xhr.data.code == 0) {
141     $scope.isSame2 = false;
142     $scope.pswError = false;
143     $cookieStore.put('token', xhr.data.data.token);
144     $cookieStore.put('hiddenLogin', {
145       status: true,
146       date: dateToday()
147     }); // 是否保存cookie登录, status=true表示已登录人此次登录
148     $scope.data = xhr.data.data;
149     $scope.data.token = encodeURI($scope.data.token);
150     if (redirectUrl) {
151       var url = "";
152       if (redirectUrl.indexOf('?') > 0)
153         url = redirectUrl + '&state=' + state + '&token=' + $scope.data.token;
154       else
155         url = redirectUrl + '?state=' + state + '&token=' + $scope.data.token;
156       window.location.href = url;
157     } else if (backUrl) {
158       location.href = backUrl;
159     } else {
160       window.location.href = "/";
161     }
162   } else if (xhr.data.code == 1003) {
163     $scope.pswError = true;
164   } else if (xhr.data.code == 1002) {
165     $scope.isSame2 = true;
166   } else {
167     alert(xhr.data.message);
168   }
169 });
170 $scope.enter = function () {

```

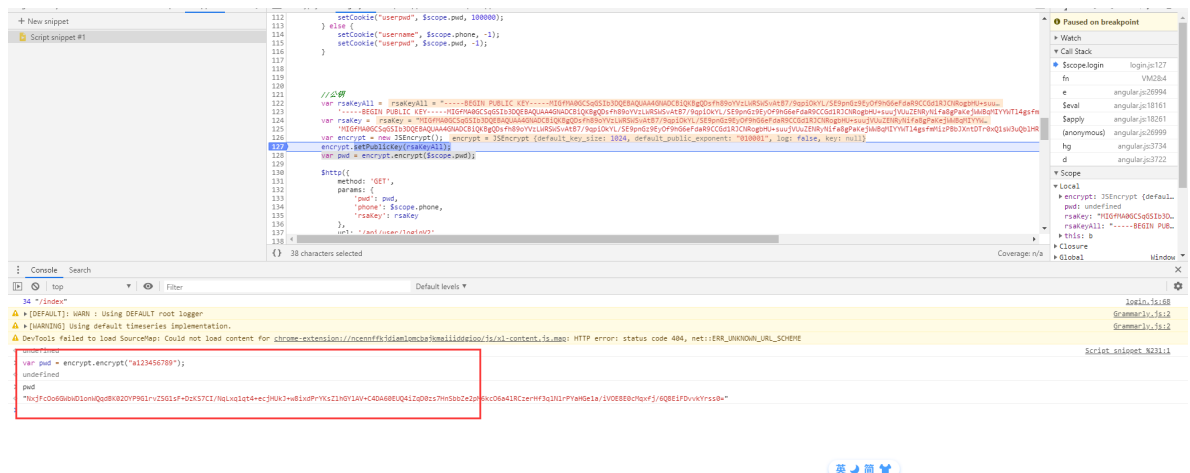
鼠标放到这行上（注意断点的位置）就可以看到蓝色可点击 直接点一下 就跟到这个对象中去了 接着把里边的代码全都复制下来



这便是这个加密方法 这里我就不复制出来了 因为太大了 这样的话 影响阅读 大家把他复制下来
还记得之前的 新建一个JS脚本吗？



把内容粘贴进去 右下角“Ctrl + Enter”运行脚本



这样便获得了加密的内容~

最后大家可以按照自己的需求 搭建服务 模拟登陆了！