四叶草网络安全学院第一届ctf比赛题目WP

原创

kerve 于 2021-02-26 11:23:02 发布 1422 收藏 2

分类专栏： CTF 文章标签： ctf 四叶草网络安全学院 抚琴

抚琴

大家好关于这次比赛密码学和部分移动安全如下

没错在这里我们来解题吧（不要吐槽我，嘴下饶人）

1

```
某加密算法实现如下
import java.nio.charset.Charset;
public class DeEnCode {
    private static final String key0 = "2021.2.26";
    private static final Charset charset = Charset.forName("UTF-8");
    private static byte[] keyBytes = key0.getBytes(charset);
    public static String encode(String enc){
        byte[] b = enc.getBytes(charset);
        for(int i=0,size=b.length;i<size;i++){
            for(byte keyBytes0:keyBytes){
                b[i] = (byte) (b[i]^keyBytes0);
            }
        }
        return new String(b);
    }
加密flag后为：Q[VPLDRTwQBF^YJ
写出解密算法求出flag
```

解：

```
mport java.nio.charset.Charset;
public class DeEnCode {

    private static final String key0 = "2021.2.26";
    private static final Charset charset = Charset.forName("UTF-8");
    private static byte[] keyBytes = key0.getBytes(charset);


    public static String decode(String dec){
        byte[] e = dec.getBytes(charset);
        byte[] dee = e;
        for(int i=0,size=e.length;i<size;i++){
            for(byte keyBytes0:keyBytes){
                e[i] = (byte) (dee[i]^keyBytes0);
            }
        }
        return new String(e);
    }

    public static void main(String[] args) {
         String dec = decode("Q[VPLDRTwQBF^YJ")
;
        System.out.println(enc);
        System.out.println(dec);
    }
}

Flag:flag{sec@fuqin}
```

2、凯撒大帝用MD5三步跨栏套娃

```
R000VEdOUlRHTTNETU5SV0dNMlRHTlJUR1VaVENOUlVHTTRER05aV0dRM0RNTVpTR00zREdOSldHNFpUSU1aWEdNMkRNTlpUSEFaVEtOUlhHTTRU
TU9KVEdRWlRFTVpWR000VEdNST0=
```

解：

```
打开题目
R000VEdOUlRHTTNETU5SV0dNMlRHTlJUR1VaVENOUlVHTTRER05aV0dRM0RNTVpTR00zREdOSldHNFpUSU1aWEdNMkRNTlpUSEFaVEtOUlhHTTRU
TU9KVEdRWlRFTVpWR000VEdNST0=
题目为套娃猜想可能与base家族有关
https://www.qqxiuzi.cn/bianma/base64.htm

GM4TGNRTGM3DMNRWGM2TGNRTGUZTCNRUGM4DGNZWGQ3DMMZSGM3DGNJWG4ZTIMZXGM2DMNZTHAZTKNRXGM4TMOJTGQZTEMZVGM4TGMI=

3936336666353635316438376466323635673437346738356739693432353931

963ff5651d87df265g474g85g9i42591
题目提示凯撒三步则推测凯撒加密位移为3
https://www.qqxiuzi.cn/bianma/kaisamima.php
963cc5651a87ac265d474d85d9f42591
题目提示有MD5则在线解密
https://www.cmd5.com/

Flag: flag{sec2021}
```

3、另类rsa

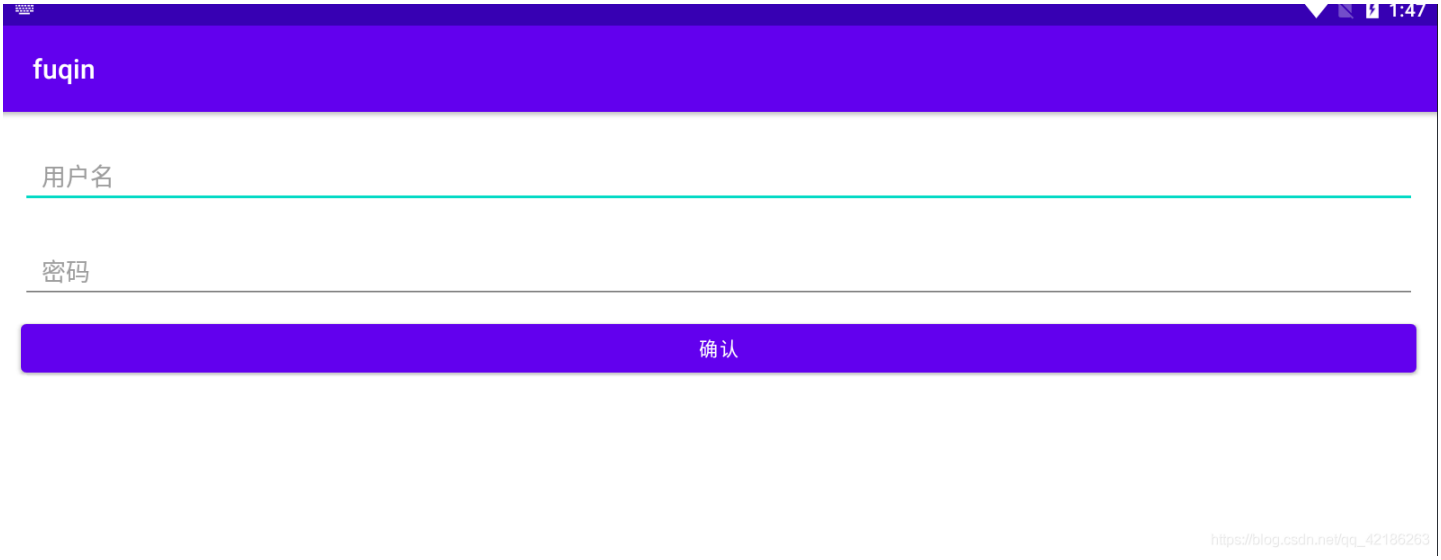
在RSA体制中，某给定用户的公钥e=31，n=3599，那么给该用户的私钥等于多少？
解法如下：
1.根据n=3599可得p，q分别为59和61
2.根据pq的值可推出φ(n)=5961=3480
3.ed≡1(modφ(n))等价于ed–kφ(n)=1
因此d31-3480*y=1

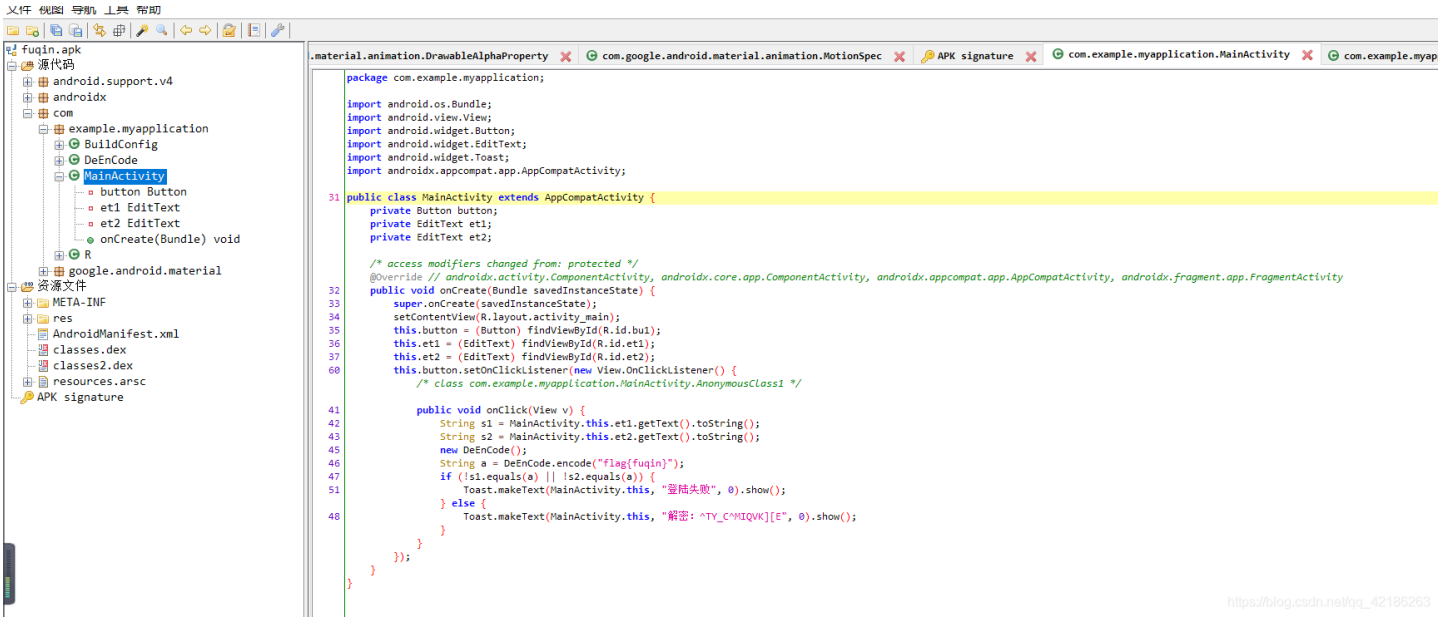
根据扩展欧几里得算法（辗转相除法）求解：
1式：3480=31*112+8 => 8=3480+(-112)31
2式：31=83+7 => 7=31+(-3)8
3式：8=71+1 => 1=8+(-1)*7

将2代入3：1=8+31+8*(-3)
1=31(-1)+84
将1代入： 1=31(-1)+[3480+31*(-112)]4
1=31(-449)+3480*4
由此得d=-449，y=4

由于d一般取正整数，所以d=d+kφ(n)=-449+1*3480=3031
所以给用户的私钥为（3599,3031）
Flag为：flag{3031}

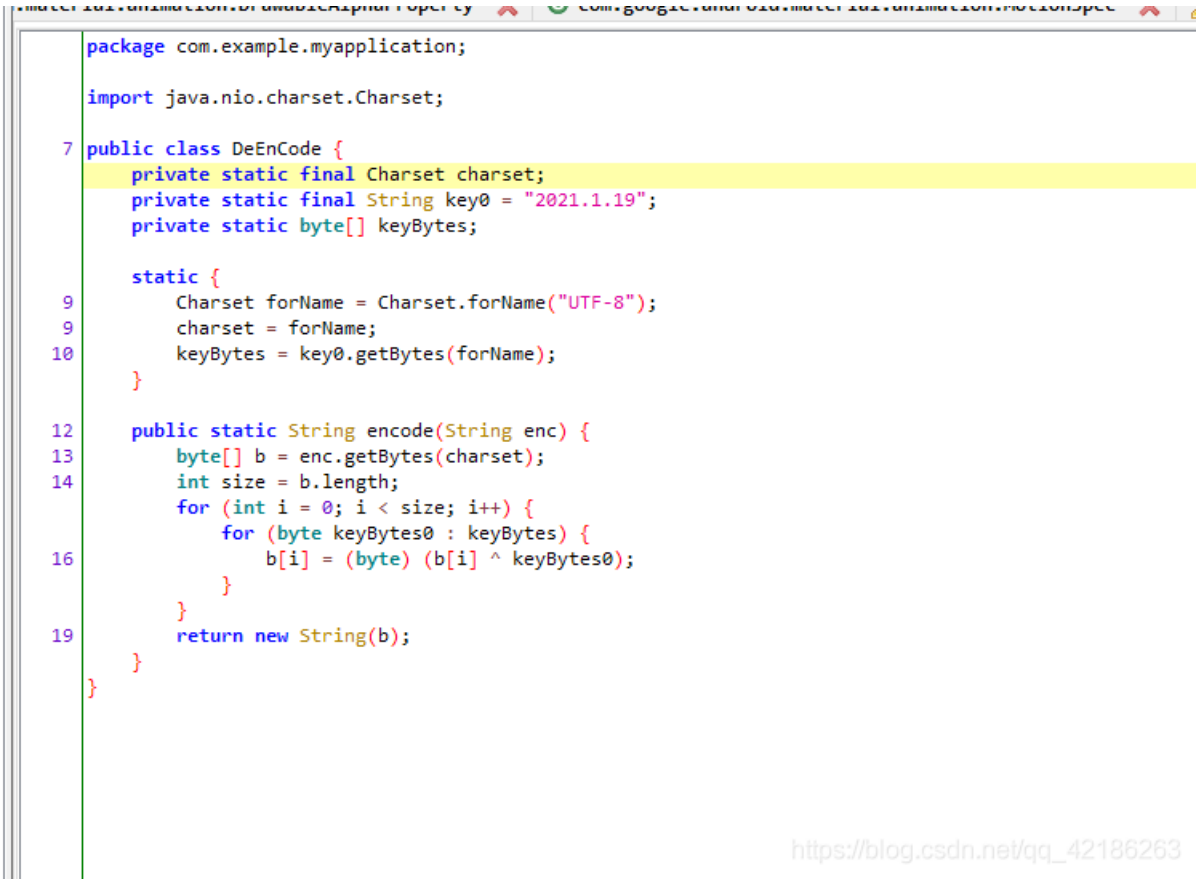
4、移动安全





```java
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private Button button;
    private EditText et1;
    private EditText et2;

    /* access modifiers changed from: protected */
    @Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, androidx.appcompat.ap
p.AppCompatActivity, androidx.fragment.app.FragmentActivity
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.button = (Button) findViewById(R.id.bu1);
        this.et1 = (EditText) findViewById(R.id.et1);
        this.et2 = (EditText) findViewById(R.id.et2);
        this.button.setOnClickListener(new View.OnClickListener() {
            /* class com.example.myapplication.MainActivity.AnonymousClass1 */

            public void onClick(View v) {
                String s1 = MainActivity.this.et1.getText().toString();
                String s2 = MainActivity.this.et2.getText().toString();
                new DeEnCode();
                String a = DeEnCode.encode("flag{fuqin}");
                if (!s1.equals(a) || !s2.equals(a)) {
                    Toast.makeText(MainActivity.this, "登陆失败", 0).show();
                } else {
                    Toast.makeText(MainActivity.this, "解密：^TY_C^MIQVK][E", 0).show();
                }
            }
        });
    }
}
```

可以看到有个flag提交了是错的再看用了DeEnCode类里面的方法，我们查看这个源代码



```java
package com.example.myapplication;

import java.nio.charset.Charset;

public class DeEnCode {
    private static final Charset charset;
    private static final String key0 = "2021.1.19";
    private static byte[] keyBytes;

    static {
        Charset forName = Charset.forName("UTF-8");
        charset = forName;
        keyBytes = key0.getBytes(forName);
    }

    public static String encode(String enc) {
        byte[] b = enc.getBytes(charset);
        int size = b.length;
        for (int i = 0; i < size; i++) {
            for (byte keyBytes0 : keyBytes) {
                b[i] = (byte) (b[i] ^ keyBytes0);
            }
        }
        return new String(b);
    }
}
```



```java
package com.example.myapplication;

import java.nio.charset.Charset;

public class DeEnCode {
    private static final Charset charset;
    private static final String key0 = "2021.1.19";
    private static byte[] keyBytes;

    static {
        Charset forName = Charset.forName("UTF-8");
        charset = forName;
        keyBytes = key0.getBytes(forName);
    }

    public static String encode(String enc) {
        byte[] b = enc.getBytes(charset);
        int size = b.length;
        for (int i = 0; i < size; i++) {
            for (byte keyBytes0 : keyBytes) {
                b[i] = (byte) (b[i] ^ keyBytes0);
            }
        }
        return new String(b);
    }
}
```


可以看到这是加密算法分析一下啊 将参数转换为字节存在数组b里面 然后遍历这个数组并且和keyBytes进行异或 应该需要对这个flag{fuqin}进行加密试着去加密

```java
import java.nio.charset.Charset;


public class DeEnCode {

    private static final String key0 = "2021.1.19";
    private static final Charset charset = Charset.forName("UTF-8");
    private static byte[] keyBytes = key0.getBytes(charset);

    public static String encode(String enc){
        byte[] b = enc.getBytes(charset);
        for(int i=0,size=b.length;i<size;i++){
            for(byte keyBytes0:keyBytes){
                b[i] = (byte) (b[i]^keyBytes0);
            }
        }
        return new String(b);
    }
 public static void main(String[] args) {
        String s="flag{fuqin}";
        String enc = encode(s);
        System.out.println(enc);
    }
}
```

运行一下

```
E:\jdk\bin\java.exe "
^TY_C^MIQVE
```

弹出来一串字符串让我们解密这个东西回想当时的加密算法写出解密算法解这个字符串

```
import java.nio.charset.Charset;


public class DeEnCode {

    private static final String key0 = "2021.1.19";
    private static final Charset charset = Charset.forName("UTF-8");
    private static byte[] keyBytes = key0.getBytes(charset);
public static String decode(String dec){
    byte[] e = dec.getBytes(charset);
    byte[] dee = e;
    for(int i=0,size=e.length;i<size;i++){
        for(byte keyBytes0:keyBytes){
            e[i] = (byte) (dee[i]^keyBytes0);
        }
    }
    return new String(e);
}
 public static void main(String[] args) {
        String dec = decode("^TY_C^MIQVK][E");
        System.out.println(dec);
    }
}
```

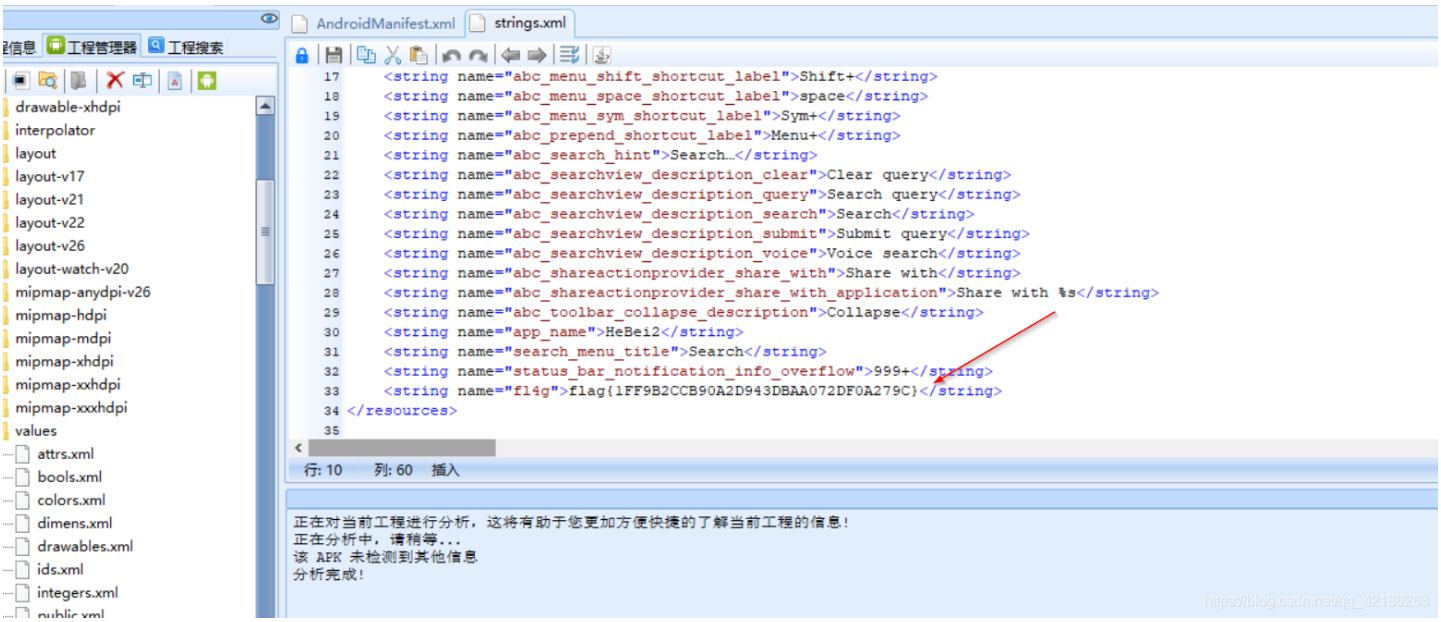
运行一下：

```
flag{fuqinsec}

Process finished with exit code 0
```

flag{fuqinsec}

5、安卓1

不要脱壳



6、rsa

脚本我放这里

```
import binascii
import sys
sys.setrecursionlimit(1000000)
def ByteToHex(bins):
    return ''.join(["%02X" % x for x in bins]).strip()
def n2s(num):
    t = hex(num)[2:-1]  # python
    if len(t) % 2 == 1:
        t = '0' + t
    #print(t)
    return(binascii.a2b_hex(t).decode('latin1'))
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        print('modular inverse does not exist')
        return 'null'
    else:
        return x % m
c = 38230991316229399651823567590692301060044620412191737764632384680546256228451518238842965221394711848337832459443844446889468362154188214840736744657885858943810177675871991111466653158257191139605699916347308294995664530280816850482740530602254559123759121106338359220242637775919026933563326069449424391192
p = 28805791771260259486856902729020438686670354441296247148207862836064657849735343618207098163901787287368569768472521344635567334299356760080507454640207003
q = 15991846970993213322072626901560749932686325766403404864023341810735319249066370916090640926219079368845510444031400322229147771682961132420481897362843199
e = 35461110244130757205657218182792589919834535022875373093108939327546391654445662689424541509610783446577840953237318712531855461472259930179152891621283936812106603554100880826153450058602365276771227162578520428096468800468032830012484968047710530251937737009257810782711682139182621097232037761496754782761 9
n = p * q
d = modinv(e, (p - 1) * (q - 1))
m = pow(c, d, n)
print (m)
```

这次出题主要是引领小白来学习安全的，各位大佬不要吐槽题目简单。后面有机会会出点难题