[Cryptsec](#)

Cryptography and security

HOW TO

My Write-up on BCTF 2016: Special RSA (Crypto 200)

- 21/03/201612/11/2016
- YANAPERMANA
- LEAVE A COMMENT

Problem

Given problem as follows:

```
While studying and learning RSA, I knew a new form of encryption/decryption wit

I encrypted msg.txt and got msg.enc as an example for you.

$ python special_rsa.py enc msg.txt msg.enc

Can you recover flag.txt from flag.enc?

special_rsa.zip.f6e85b8922b0016d64b1d006529819de
```

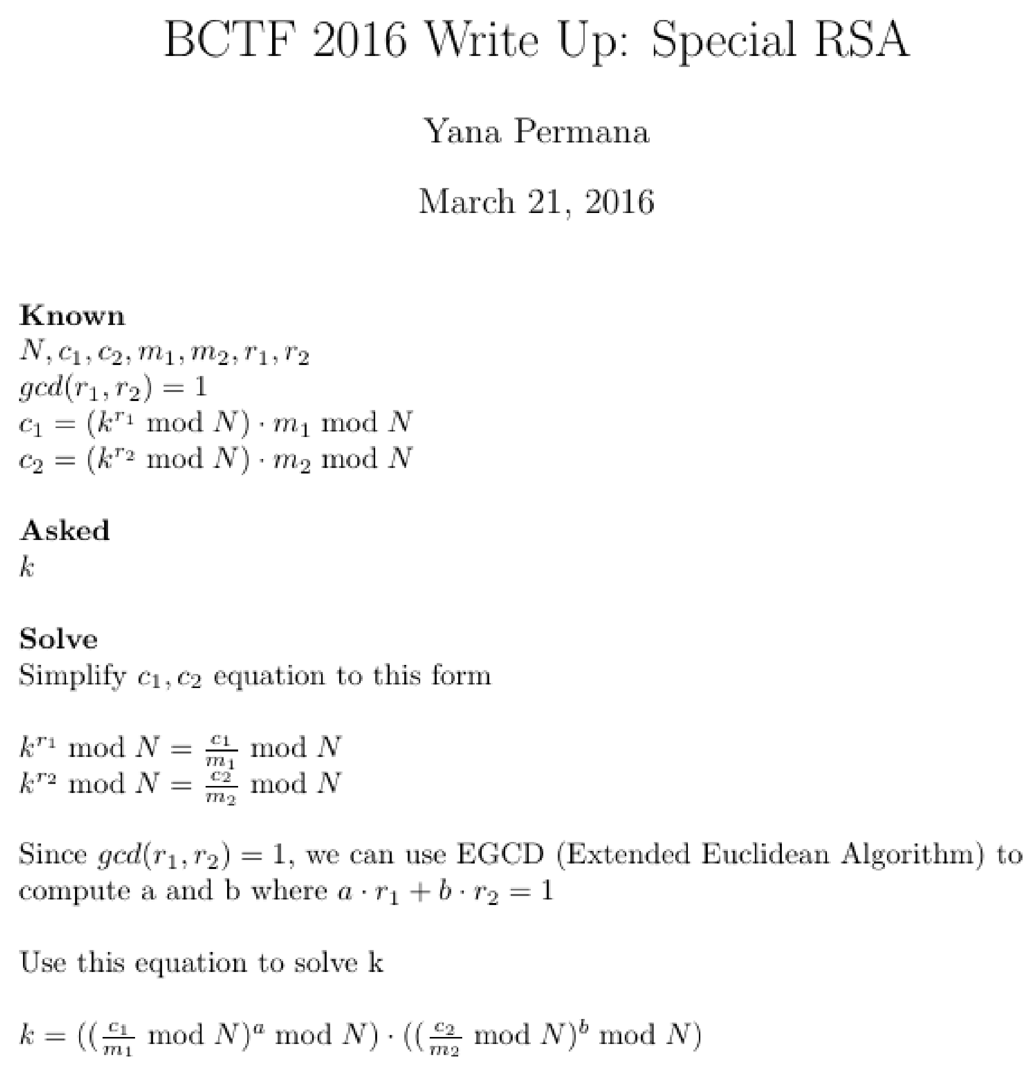
Completion

Given a special_rsa.zip.f6e85b8922b0016d64b1d006529819de (https://github.com/yanapermana/ctf-2016/tree/master/bctf/crypto/special-rsa) file that contains:

```
flag.enc
special_rsa.py
msg.enc
msg.txt
```

The purpose of this challenge is to decrypt flag.enc files with hidden key.

After reading the encryption and decryption process on special_rsa.py file, I made simple equation to find the hidden key.



BCTF 2016 Write Up: Special RSA

Yana Permana

March 21, 2016

Known
$N, c_1, c_2, m_1, m_2, r_1, r_2$
$gcd(r_1, r_2) = 1$
$c_1 = (k^{r_1} \bmod N) \cdot m_1 \bmod N$
$c_2 = (k^{r_2} \bmod N) \cdot m_2 \bmod N$

Asked
k

Solve
Simplify c_1, c_2 equation to this form

$k^{r_1} \bmod N = \frac{c_1}{m_1} \bmod N$
$k^{r_2} \bmod N = \frac{c2}{m_2} \bmod N$

Since $gcd(r_1, r_2) = 1$, we can use EGCD (Extended Euclidean Algorithm) to compute a and b where $a \cdot r_1 + b \cdot r_2 = 1$

Use this equation to solve k

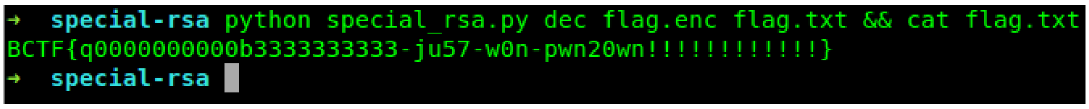
$k = ((\frac{c_1}{m_1} \bmod N)^a \bmod N) \cdot ((\frac{c_2}{m_2} \bmod N)^b \bmod N)$

Here it is solv.sage (https://github.com/yanapermana/ctf-2016/blob/master/bctf/crypto/special-rsa/solv.sage)

```
N = 23927411014020695772934916764953661641310148480977056645255098192491740356525

c1 = 14548997380897265239778884825381301109965518989661808090688952232381091726
c2 = 12793942795110038319724531875568693507469327176085954164034728727511164833

m1 = 82460741826420911255783118283748436989942332438113476912293348292187007286
m2 = 15575051453858521753108462063723750986386093067763948316612157946190835527

r1 = 12900676191620430360427117641859547516838813596331616166760756921115466932
r2 = 77189751594023896179245431001139675122801316302866240781023681661854434662

_, a, b = xgcd(r1, r2)
k = pow((c1/m1 % N), a, N) * pow((c2/m2 % N), b, N)
print k
```

```
The key is 175971776542095822590595405274258668271271366360140578776612582276966
```

After got the key, we can decrypt flag.enc file easily.



```
→  special-rsa python special_rsa.py dec flag.enc flag.txt && cat flag.txt
BCTF{q0000000000b3333333333-ju57-w0n-pwn20wn!!!!!!!!!!!!}
→  special-rsa
```

Flag: **BCTF{q0000000000b3333333333-ju57-w0n-pwn20wn!!!!!!!!!!!!}**

Advertisements