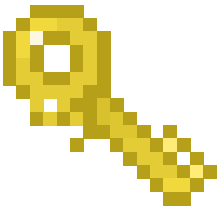
Am I doing it right?



Type: Cryptography

Difficulty: Medium

Flag: `TFCCTF{1_w0n7_4dm1t_h0w_l0ng_th1s_t00k_4_me...congrats,tho!}`

Points: 188

Description

Am I doing this RSA encryption right?

Solution

We get a public key, that was used to encrypt a message. I used https://www.dcode.fr/rsa-cipher to see if we can get the values for p,q,d, and it turns out successfull because p and q are equal! (big no-no!)

With these values I can replicate the private key, using a Go script I created:

```
func main() {
  var primes []*big.Int
  a := new(big.Int)
  n := new(big.Int)
  d := new(big.Int)

  p, _ = a.SetString("133150398268195275743440564494273922289580211854275732500110288734747827954102678337833707292013932698242219254 6
  n, _ = n.SetString("177290285589790194858464200346146017818552868857721160331159982891306632187932491351030979414066155947835644870 5
  d, _ = d.SetString("350025024832734994167213679033032229817394078177221126008953571056138748169684072446250185794070828968223356116 4

  primes = append(primes, p)
  primes = append(primes, p)

  privKey := new(rsa.PrivateKey)
  privKey = &rsa.PrivateKey{
    PublicKey: rsa.PublicKey{
      N: n,
      E: 65537,
    },
    D:           d,
    Primes:      primes,
    Precomputed: rsa.PrecomputedValues{},
  }

  privKeyTemp, _ := rsa.GenerateKey(rand.Reader, 2048)
  privKeyTemp.Primes = primes
  privKeyTemp.N = n
  privKeyTemp.D = d
  privKeyTemp.Precompute()
  publicKey := privKeyTemp.PublicKey
  privKey.Precompute()

  var privateKeyBytes = x509.MarshalPKCS1PrivateKey(privKeyTemp)
  privateKeyBlock := &pem.Block{
    Type:  "RSA PRIVATE KEY",
    Bytes: privateKeyBytes,
  }

  privatePem, _ := os.Create("private.pem")
  _ = pem.Encode(privatePem, privateKeyBlock)
}
```

With that done, we can simply decrypt the encrypted file using

```
openssl rsautl -decrypt -inkey private.pem -in chall.enc > chall.txt
```

And we get the flag!