Cryptography
Day 02

Andrew Hou, Richard Li

October 2, 2021

Overview

- Intro + Review
  - Public Key Cryptography
  - Modular Arithmetic
- RSA
  - RSA Definition
  - N, e, and d
- Basic RSA Exploits
  - Exploit 1: e Too Small
  - Exploit 2: Primes Too Small
  - Exploit 3: e Too Small... Again (Håstad's broadcast attack)
- ...And Beyond

Cryptohack Docker

https://github.com/cryptohack/cryptohack-docker

Public Key Cryptography

- Yesterday: mostly private key cryptography
- RSA is public key cryptography
- Public key cryptography relies on "hard" problems
- Public-private key pair

Modular Arithmetic

- Yesterday: addition modulo N
- Today: multiplication/exponentiation modulo N
- **multiplicative inverse** of a mod N: $a \cdot a^{-1} \equiv 1 \pmod{N}$
- **Euler's totient function**: $\varphi(N)$
- **Euler's theorem**: $a^{\varphi(N)} \equiv 1 \pmod{N}$

RSA Definition

- Start with two hidden primes p and q. $N = p \cdot q$.
- Let m be the message, and (e, N) be the public key. Then encrypt the message to get ciphertext $c \equiv m^e \pmod{N}$.
- The private key $d \equiv e^{-1} \pmod{\varphi(N)}$ decrypts the message using the same method. $m \equiv c^d \equiv (m^e)^d \equiv m^1 \pmod{N}$.

N, e, and d

- When N is the product of two primes p and q, $\varphi(N) = (p-1)(q-1)$.
- Reminder that $d \equiv e^{-1} \pmod{\varphi(N)}$... which would be easy to calculate if we know $\varphi(N)$.
- But to calculate $\varphi(N)$ we would have to know the factors of N, which is hard.
- Also note that for $d \equiv e^{-1}$ to exist, e must be relatively prime to $\varphi(N)$.

Practice!

github.com/andrewfhou/bootcamp-2021-crypto/

Challenges under day2/

Exploit 1: *e* Too Small

```
pin = bytes_to_long(b'4321')

e = 3
p = getPrime(512)
q = getPrime(512)
phi = (p-1)*(q-1)
d = inverse_mod(e,phi)
N = p*q

print('Can you recover my 4-digit bank PIN')
print(f'N: {N}\ne: {e}')
C = pow(pin,e,N)
print(f'C: {C}')
```

Exploit 1: e Too Small

- We're told that the PIN is 4 digits, which means the maximum is 9999
- Remember that $m^e \equiv c \mod N$
- What happens when $m^e < N$?
- We can take $\sqrt[3]{c} = m$

Exploit 2: Primes Too Small

```
m = bytes_to_long(b'flag{primes!}')
e = 0x10001 #65537

p = getPrime(64)
q = getPrime(64)
r = getPrime(64)
s = getPrime(64)
phi = (p-1)*(q-1)*(r-1)*(s-1)
N = p*q*r*s
d = inverse_mod(e,phi)

c = pow(m,e,N)
print(f'N: {N}\ne: {e}\nC: {c}')
```

Exploit 2: Primes Too Small

- Those primes look awfully small...
- $\varphi(N) = \varphi(p)\varphi(q)\varphi(r)\varphi(s)$
- We can now compute $d \equiv e^{-1} (\bmod \varphi(N))$
- Now that we know d, decryption is trivial

Chinese Remainder Theorem

Let $n_1, n_2, \ldots, n_k$ be integers greater than 0, and *pairwise coprime* - that is, the GCD of all n_i is 1

Let $a_1, a_2, \ldots, a_n$ be arbitrary integers. Then,

$$
\begin{aligned}
x &\equiv a_1 \mod n_1 \\
x &\equiv a_2 \mod n_2 \\
&\ldots \\
x &\equiv a_n \mod n_n
\end{aligned}
$$

Has a solution, and any two solutions are are congruent modulo N. That is to say for two solutions x_1, x_2, we have $x_1 \equiv x_2 \mod N$

Håstad's broadcast attack

- Suppose the same message m is sent to a number of people using the same small public exponent, lets say $e = 3$
- Suppose we intercept c_1, c_2, c_3 where $c_i \equiv m \mod n_i$
- Where have we seen a system of modular congruences?
- Using CRT, we can compute $c \equiv m^3 \mod n_1 n_2 n_3$
- Since $m < n_i$, it follows that $m^3 < n_1 n_2 n_3$, so $c \equiv m^3$

Exploit 3: *e* Too Small... Again

```
moduli = []
for _ in range(3):
    p = getPrime(128)
    q = getPrime(128)
    N = p*q
    moduli.append(N)

e = 3
m = bytes_to_long(b'Here\'s my secret message')
ciphertexts = []
for n in moduli:
    ciphertexts.append(pow(m,e,n))

print(ciphertexts)
print(moduli)
```

What's Next?

Modern Cryptography

- AES
- Elliptic Curve Cryptography (ECC)
- Post-Quantum Cryptography
  - Lattice Crypto
  - LWE
  - Multivariate Crypto

How do I learn more?

- Cryptohack is an always-on crypto focused CTF that builds from nothing all the way to ECC!
- cryptohack.org
- cryptopals.com is another excellent set of crypto problems