Project 3: Crypto – Have fun with RSA

March 1, 2017

1 Intro – RSA

RSA is one of the widely used public key cryptosystem in real world. It's composed of three algorithms: key generation (**Gen**), encryption (**Enc**), and decryption (**Dec**). In RSA, the public key is a pair of integers (e, N), and the private key is an integer d.

Gen The key pair is generated by the following steps:

1. Choose two distinct big prime numbers with the same bit size, say p and q.
2. Let $N = p * q$, and $\phi(N) = (p-1) * (q-1)$.
3. Pick up an integer e, such that $1 < e < \phi(N)$ and $gcd(e, \phi(N)) = 1$.
4. Let $d \equiv e^{-1} \mod \phi(N)$.
5. Return (N, e) as public key, and d as private key.

Enc To encrypt integer m with public key (N, e), the cipher integer $c \equiv m^e \mod N$.

Dec To decrypt cipher integer c with private key d, the plain integer $m \equiv c^d \mod N$.

2 Task1 – Get familiar with RSA (10 points)

The goal of this task is to make you familiar with RSA.

You're given an RSA key pair (N, e) and d, and an unique cipher text c. You're required to get your own unique message m. Each student's key pair and cipher text can be found in *"keys.json"*. You're required to implement the function `rsa(c, e, d, N)` in "rsa.py" to get the plain text. Please submit the plain text output from the rsa.py.

```
def rsa(c, e, d, N):

    """
    :param c: The cipher text
    :param N,e: The public key pair
    :param d: The private key
    :return: The number representing the plaintext
    """
    m = None

    # your code starts here

    # your code ends here

    return m
```

3 Task2 – Attack small key space (30 points)

In real world, the commonly used RSA key size if 1024 bits, which is hard for attackers to traversal the whole key space with limited resources. Now, you're given an unique RSA public key, of which the key size is pretty small (64 bits), your goal is to get the private key. All keys can be found in *"keys.json"*.

You're required to write some code in *"get_pri_key.py"* to get the private key:

- TODO1: implement function `get_factors`, n is the given public key (64 bits), your goal is to get its factors. (You can cheat on this subtask, as long as you can get the right p and q.)

```
def get_factors(n):
    p = 0
    q = 0

    # your code starts here

    # your code ends here
    return (p, q)
```

- TODO2: implement function `get_key` to get the private key.

```
def get_key(p, q, e):
```

```
    d = 0

    # your code starts here

    # your code ends here
    return d
```

You're required to submit: (1) your unique private key, in hex format; (2) the *"get_pri_key.py"* file; (3) a brief description about your steps to get the private key.

4 Task3 – Where is Waldo? (60)

Read the paper "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices".

You're given an unique RSA public key, the RNG (random number generator) used in the key generation is vulnerable. Also, all your classmates's public keys are generated by the same RNG on the same system. Your goal is to get your unique private key. All keys can be found in *"keys.json"*.

You're required to complete some code in *"find_waldo.py"* to get the private key:

- TODO1: implement function `is_waldo`, $n1$ is your own key, $n2$ is one of your classmate's key, try to find out whether this classmate is Waldo.

```
def is_waldo(n1, n2):
    result = False

    #your code start here

    #your code ends here

    return result
```

- TODO2: since you've successfully found your Waldo among your classmates, now you have to implement function `get_private_key` to get your own unique private key. $n1$ is your public key, $n2$ is Waldo's public key.

```
def get_private_key(n1, n2, e):
    d = 0

    #your code starts here
```

```
    #your code ends here

    return d
```

You're required to submit: (1) your unique private key, in hex format; (2) your classmate's (Waldo) name; (3) the *"find_waldo.py"* file; (4) your understanding about the weak key problem caused by Ps and Qs; (5) a simple description about your steps to get the private key.

5 Submission

Note that all students' keys are different, so don't copy and paste answers from your classmates. In total, please submission the following files:

1. "submission.txt": the first line is the plain text from task1; the second line is the private key from task2; the third line is the private key from task3; the forth line is the Waldo.
2. "get_pri_key.py" and "find_waldo.py".
3. "writeup.pdf": a brief explanation about how you get the private key in task2; your understanding about the weak key problems in task3; a brief explanation about how you get the private key in task3.