COMP 3355 Cyber Security

Cryptography Tasks- Assignment 1

Submitted by: BHATIA Divtej Singh

UID: 3035832438

Task 1

1.1 How does RSA encrypt a message?

RSA is an asymmetric encryption technique that involves creating two keys, one for encryption and the other for decryption. To initiate this process, RSA generates a pair of keys. This involves selecting two prime numbers, *P* and *Q*, and computing their product (*N*).

Then, find the totient, $\phi(N)$,

$$\phi(N) = (P - 1) * (Q - 1).$$

Such that, $1 < e < \phi(N)$, and

e is co-prime with both $\phi(N)$ and N.

Once the values of e and N have been determined, they constitute the public key, represented as (e, N). This public key is then utilized to encrypt a message using the following formula:

$$C = M^{e} \bmod N$$

In this formula, C represents the encrypted message, and M is the message that is to be encrypted.

1.2 How can we decrypt the message if we have the private key?

The private key consisting of two values: **d** (the decryption exponent) and **N** (the modulus). **C** is the ciphertext to be decrypted. Then,

Using the decryption formula:

$$M = C^{d} \bmod N$$

Decrypted Message / Original message M can be retrieved , which was encrypted using the corresponding public key.

Task 2

2.1 How do you get the factors p and q from the given public key?

In the task, the given unique RSA public key has a fairly small key length (64bits).

The approach used to get p and q:

- Iterates through numbers from 2 to the square root of **N** (inclusive) to find a factor **i** of **N**.
- For each factor **i** found, it checks if both **i** and **N // i** are prime using the '**is_prime**' function.
- The '**is_prime**' function checks if a number is prime or not. It starts by checking for specific cases (<= 3) and then uses a more efficient prime-checking algorithm for larger numbers.
- If both **i** and **N // i** are prime, it returns them as **p** and **q**, which are the prime factors of **N**.
- If no prime factors are found, it returns **0** for both **p** and **q**.

2.2 How do you generate the relative private key from the factors?

The private key in RSA consists of N and d. N is the product of p and q, and d is calculated as follows:

- First, $\boldsymbol{\phi} = (\boldsymbol{P} - \mathbf{1}) * (\boldsymbol{Q} - \mathbf{1})$, which is the totient of N.
- d is calculated using the function for modular multiplicative inverse

  d = pow(e, -1, $\boldsymbol{\phi}$),

  which calculates e^(-1) mod $\boldsymbol{\phi}$ to find the decryption exponent d.

** Note that it is given that: $pow(a, b, c)$, the result of $a^b\%c$. Although the result is the same as $(a ** b)\%c$, it is strongly recommended to use $pow(a, b, c)$ for better performance.

Task 3

3.1 What is the workflow of Ps and Qs attack?

Under the Ps and Qs attack, two or more RSA moduli (N values) share one prime factor (P) but have different second prime factors (Q). The attacker can factorize these moduli and potentially recover the private keys.

This is the workflow of the Ps and Qs attack:

- A set of RSA public keys, each consisting of a modulus N and a public exponent e. If the keys are generated using a vulnerable RNG, then the likelihood of sharing common factors increases.
- Then, we can calculate the GCD of the moduli N of all pairs of public keys. If the GCD of two moduli is greater than 1, it indicates that these moduli share at least one common prime factor (P).
- Once a pair of moduli with a common P factor is found, the attack has revealed P. P is common to both RSA keys and once P is known, the corresponding Q values for each modulus can be found as explained in the next point.
- To find the Q values for each modulus, divide the modulus (N) by the known P factor to obtain Q. Two different Q values are obtained, one for each modulus
- Now that we have the values of P, Q1, and Q2, the value of the private keys (d1 and d2) for the RSA keys can be computed using:

$$\mathbf{d = pow(e, -1, \phi)\ , where\ } \phi = (P - 1) * (Q - 1)$$

Then, any messages can be decrypted between the two end- users. The attacker can also intercept and alter encrypted communications or impersonate the users.

3.2 How can we prevent Ps and Qs attack?

Ps and Qs attacks can be prevented to a certain extent by using secure random number generators. Weak RNGs is a vulnerability that can lead to shared prime factors, which makes Ps and Qs attacks possible. For enhanced security, it is important to examine keys for repetition and factorability.

The primary concern with Ps and Qs attacks is the accidental reuse of prime factors. But in some cases, following proper key generation procedures to ensure unique prime factors for each key pair and regularly rotating can prevent these attacks

---

Task 4 on next page

Task 4

4.1 What is the workflow of the broadcast attack ?

A broadcast attack uses a common modulus in multiple RSA public keys to recover the corresponding private keys. When different parties have the same modulus for their RSA encryption, then those parties are vulnerable to this kind of attack.

The broadcast attack involves calculating modular inverses, using the **Chinese Remainder Theorem** to combine information from multiple ciphertexts and moduli, and finally taking the cubic root to recover the original plaintext message.

This is the workflow of the broadcast attack:

1. Calculate the modular inverses:
    - Calculate x1, which is the modular inverse of N1 modulo (N2 * N3).
    - Calculate x2, which is the modular inverse of N2 modulo (N3 * N1).
    - Calculate x3, which is the modular inverse of N3 modulo (N1 * N2).
2. Calculate the Chinese Remainder Theorem (CRT) result (S):
    - Use the modular inverses (x1, x2, x3) to calculate W. W is a weighted sum of the ciphertexts (C1, C2, C3) and moduli (N1, N2, N3).
    - The formula used to calculate W is:

      W = (x1 * C1 * N2 * N3) + (x2 * C2 * N1 * N3) + (x3 * C3 * N1 * N2).
3. Calculate the cubic root of W modulo (N1 * N2 * N3):
    - Take the cubic root of W modulo (N1 * N2 * N3). The cube root is used because the original message was encrypted three times, and taking the cube root helps recover the original message.

      m = root3(W % (N1 * N2 * N3))

4.2 Can we recover the message with two ciphertexts instead of using three?

It is not possible to recover the original message with 2 ciphertexts in an RSA broadcast attack. The is related to mathematical methods, which are used to solve for the private key components.

When we have only two ciphertexts (C1 and C2) and two public keys (N1 and N2), there is not enough information to effectively apply the CRT to recover the private key components. In the RSA encryption process, at least two unique prime factors (p and q) of the modulus (N) are needed to recover the private key.

Therefore, there is a large possibility of missing coefficients, hence the RSA broadcast attack typically requires three or more ciphertexts with the same modulus to effectively recover the private keys.

However, consider the special case: We can recover the private key components with just two ciphertexts if they share a common factor. It's not necessary to have three or more ciphertexts under this case.

End of Submission