Tutorial 7: RSA

By the end of this section of the practical, the students should be able to:

- to understand Public Key RSA Algorithm
- to understand the generation process of public and private keys
- to relate encryption and decryption process using RSA to PKI.
- to sign and verify a digital signature

Please do this exercise in pair.

1. Overview

RSA is an algorithm for public-key cryptography. The algorithm was publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman at MIT; the letters RSA are the initials of their surnames.

Clifford Cocks, a British mathematician working for the UK intelligence agency GCHQ, described an equivalent system in an internal document in 1973, but given the relatively expensive computers needed to implement it at the time, it was mostly considered a curiosity and, as far as is publicly known, was never deployed. His discovery, however, was not revealed until 1997 due to its top-secret classification, and Rivest, Shamir, and Adleman devised RSA independently of Cocks' work. MIT was granted US patent 4405829 for a "Cryptographic communications system and method" that used the algorithm in 1983. The patent expired on 21 September 2000.

2. RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

Step 1. Choose two distinct large random prime numbers P and Q.

Step 2. Compute $N = P \cdot Q$ is used as the modulus for both the public and private keys

Step 3. Compute the totient: $\phi(N) = (P-1)(Q-1)$

Step 4. Choose an integer e such that $1 < e < \phi(N)$, and share no factors other than 1 (coprime) with N and $\phi(N)$. We need to make sure that $\gcd(e, \phi(N)) = 1$, e is released as the public key exponent

Step 5. Compute d to satisfy the congruence relation $d \cdot e \equiv 1 \bmod \phi(N)$ i.e. $d = e^{-1} \bmod \phi(N)$.

Notes on the above steps:

Step 1: Numbers can be probabilistically tested for primality.
Step 2: There is a different formula used in PKCS#1 v2.0.
Step 3: A popular choice for the public exponent is $e = 2^{16} + 1 = 65537$.
Steps 4 and 5 can be performed with the extended Euclidean algorithm.

Some applications choose smaller e values such as = 3, 5 or 13, 17, 19, 127, 257 instead. This is done to make encryption and signature verification faster on small devices like smart cards but small public exponents may lead to greater security risks.

3. Encrypting and Decrypting messages

Alice transmits her public key (e, n) to Bob and keeps the private key secret. Bob then wishes to send message **M** to Alice. He wants to compute the ciphertext C. To encrypt a message M Bob shall used the public key (e, n) and compute the ciphertext $C = M^e \bmod n$. Bob then transmits C to Alice.

Alice can recover M from C by using her private key (d, n) by the following computation: To decrypt a message the owner of the private key (d, n) will compute back $M = C^d \bmod n$. Given C, she can recover the original message **M**.

4. Exercise A using Excel or calculator.

0. Let us generate the public and private keys. Write your matrix ID in hexa decimals. Take your ID modulo 16. Take i = 11 mod 16 = $0B_{16}$	
1. Alice randomly chooses a large prime, P_A, from the set $i \in \{0, 1, 2, \ldots, 15\}$ $P_i \in$ {557, 569, 577, 593, 601, 613, 619, 641, 647, 659, 673, 683, 701, 719, 733, 743}	1. Bob randomly chooses a large prime, P_B, from the set $i \in \{0, 1, 2, \ldots, 15\}$ $P_i \in$ {563, 571, 587, 599, 607, 617, 631, 643, 653, 661, 677, 691, 709, 727, 739, 751}
2. Alice randomly chooses another large prime, Q_A, from the set $i \in \{0, 1, 2, \ldots, 15\}$ $Q_i \in$ {787, 809, 821, 827, 839, 857, 863, 881, 887, 911, 929, 941, 953, 971, 983, 997}	2. Bob randomly chooses another large prime, Q_B, from the set $i \in \{0, 1, 2, \ldots, 15\}$ $Q_i \in$ {797, 811, 823, 829, 853, 859, 877, 883, 907, 919, 937, 947, 967, 977, 991, 1009}
3. Alice choose her public exponent E_A from the set $i \in \{0, 1, 2, \ldots, 15\}$, $e_i \in$ {13, 23, 41, 73, 83, 103, 107, 113, 127, 137, 139, 151, 163, 173, 181, 193}. Alice computes an RSA modulo $N_A = P_A \cdot Q_A$ and publishes her public key as (E_A, N_A) **Note:** Make sure that $\gcd(E_A, \phi(N_A)) = 1$.	3. Bob choose his public exponent E_B from the set $i \in \{0, 1, 2, \ldots, 15\}$, $e_i \in$ {19, 29, 47, 79, 89, 109, 113, 127, 131, 149, 157, 167, 179, 191, 197, 199}. Bob computes the modulo $N_B = P_B \cdot Q_B$ and publishes her public key as (E_B, N_B). **Note:** Make sure that $\gcd(E_B, \phi(N_B)) = 1$.

i	P_i	Q_i	e_i
0	557	787	13
1	569	809	23
2	577	821	41
3	593	827	73
4	601	839	83
5	613	857	103
6	619	863	107
7	641	881	113
8	647	887	127
9	659	911	137
10	673	929	139
11	683	941	151
12	701	953	163
13	719	971	173
14	733	983	181
15	743	997	193

4. Alice computes her totient

$$\phi(N_A) = (P_A - 1)\cdot(Q_A - 1)$$

and private exponent

$$D_A = E_A^{-1} \bmod \phi(N_A).$$

Alice keeps her private key as (D_A, N_A)

i	P_i	Q_i	e_i
0	563	797	19
1	571	811	29
2	587	823	47
3	599	829	79
4	607	853	89
5	617	859	109
6	631	877	113
7	643	883	127
8	653	907	131
9	661	919	149
10	677	937	157
11	691	947	167
12	709	967	179
13	727	977	191
14	739	997	197
15	751	1009	199

4. Bob computes his totient

$$\phi(N_B) = (P_B - 1)\cdot(Q_B - 1)$$

and private exponent

$$D_B = E_B^{-1} \bmod \phi(N_B).$$

Bob keeps his private key as (D_B, N_B)

5. Alice and Bob communicate using the Public Key Cryptosystem RSA which was never transmitted over the insecure circuit.

i. Alice would like to send the message M_A her last 5 digit of MyKAD to Bob.

Compute the ciphertext C_A.

ii. Bob will decrypt the ciphertext C_A using his private key.

iii. Bob would like to reply the message M_B his last 5 digit of MyKAD to Alice. Compute the ciphertext C_B and send to Alice.

iv. Alice will decrypt the ciphertext C_A using her private key.

v. Alice is not sure whether it is coming from Bob. Alice will request a signature from Bob.

Bob will sign the message and send to Alice.

vi. Alice will verify the message signature.