**KING SAUD UNIVERSITY**

**Information Systems Department.**

**IS493 Tutorial – Project I (Defensive)- Report Template**

**Group#: 8**

**Student ID (Leader): 438104723**

**Student Name: Fahad AlHssan**

**Student ID: 438103308**

**Student Name: Abdulelah AlMajed**

**Student ID: 438104517**

**Student Name: Salman Al Shuaiby**

**Student ID: 438102816**

**Student Name: Faisal Al Gruainy**

**Instructor: Mr. Mourad Benchikh**

# **Cipher Overview**

RSA stands for Rivest Shamir Adelman, the inventors of this algorithm, that uses a public key system, the algorithm was introduced in 1978, and still it is considered as a secure algorithm. RSA relies on a mathematical field, known as the *Number Theory,* which is the study that examines the properties of integer, divisibility, modular, and Prime numbers.

There are two keys is the RSA cryptosystem, those are, e and d, where e stands for the public key, and d stands for the private key that will encrypt and decrypt the message, so the encryption and decryption are interchangeable, such that:

To encrypt the plaintext, we say:

To decrypt:

Where n is the product of p and q, the two large numbers that are decided by the sender and receiver to encrypt the message.

N = P\*Q

And Phi(n) = (p-1)(q-1)

To choose E, gcd(E, Phi(n)) must be equal to 1, E > 1, E < Phi(n)

And to choose D, D\*E mod Phi(n) must be equal to 1

# **Used Data Structure**

There is no data structure has been used, since RSA is pretty straightforward cryptosystem. A simple set of variables collected from user input will be enough to do the encryption, so the solution was hard-coded.

# **Error handling**

The encryption of RSA is based on the values of p and q, to generate a key *e*, there is some sort of rules must be applied for such purpose, so the chosen key must be in the set of numbers A that satisfies those rules, so if the number chosen doesn’t exists on A, the program will throw an error. Or if d\*e mod Phi =/= 1, the program will throw an error also.

For example, in order to choose E, the gcd(E, Phi) must be equal to 1, and E > 1, E < Phi(n), we mean by the set A, is every number that the gcd(number, Phi) = 1, so there are a limited set of numbers, and E must be one of them.

For D, it is basically D\*E mod Phi must be equals to 1. If any of those two errors happens, the program will alert the user to “enter a valid number”.

# **Code**

## **Encoding**

|  |
| --- |
| N = P \* Q;  Phi = (P - 1) \* (Q - 1);  // a function to calculate gcd  var gcd = function(a, b) {  if (b == 0) {  return a;  }  return gcd(b, a % b);  };  var Flag = false;  if (gcd(E, Phi) == 1 && E > 1 && E < Phi)  Flag = true;  if (Flag && D \* E % Phi == 1) {    // a function to calculate the exponent and modules, since if done directly the result will be infinity (because of the large exponent)  var encrypt = function (base, exponent, modulus) {  if (modulus === 1) return 0;  var result = 1;  base = base % modulus;  while (exponent > 0) {  if (exponent % 2 === 1) //odd number  result = (result \* base) % modulus;  exponent = exponent >> 1; //divide by 2  base = (base \* base) % modulus;  }  return result;  };  // encryption  Cipher = encrypt(M,E,N); |

## **Decoding**

|  |
| --- |
| // decryption  Dec = encrypt(Cipher,D,N); |

## **RSA**

#### **Your Example**

P=2, q=7, e =5, d=11

Enciphering/ Deciphering

|  |
| --- |
| Graphical user interface  Description automatically generated |

#### **Instructor’s example**

p=53 -, q=61, n=3233, ø(n)=3120 , e=37, d=253

**Plaintext**: match = 1902

Enciphering/ Deciphering

|  |
| --- |
| Graphical user interface, application  Description automatically generated |