MATH Project Idea !!

Topic: Math behind RSA Encryption

Here is how it works (very briefly):

- Two numbers are made public. Together these are called the public key.
- One of these numbers is a product **n=p*q** of two prime numbers, and the other is a number we will call **"e"** that for technical reasons needs to be relatively prime to **ϕ(n) = (p-1)*(q-1).** That is, the greatest common divisor of **e** and **ϕ(n)** must be equal to 1.
- **GCD(ϕ(n),e) = 1**

Find the Algorithm below:

1. Enter p value (prime only)

2. Enter q value (prime only)

3. Calculate $n = pq$ value

4. Calculate RSA tfotient $\phi(n) = (p-1)(q-1)$ value.

5. Find e values such that $1 < e < \phi(n)$ & Co-prime of $\phi(n)$.

6. Display all possible values of e using point no. 5

7. Select e value (from displayed) & Enter e

8. Calculate $d(e) = 1 \bmod \phi(n)$ value

9. Use Euclidean Algorithm,

   i. $\phi(n) =$ Quotient (e) + remainder

   ii. Re substitute & simplify

   iii. Make value of 1 such that to get d value

$$1 = ?(\phi(n)) - dx(e)$$

10. We get the value of d. → if d is negative, $d = d \bmod \phi(n)$

11. print p, q, n, $\phi(n)$, e, d

encryption, public key = e

private key = d

Example :

Take $p = 11$

$q = 5$

$n = pq = 55$

$\phi(n) = (p-1)(q-1) = 10 \times 4 = 40$

$1 < e < \phi(n)$ & coprime of $\phi(n)$ is

possible values = 3, 7, 9, 11, 13, 17, ...

So, user will select some e value

put $e = 7$

$d \times e \equiv 1 \bmod \phi(n)$

$d \times 7 = 1 \bmod 40$

Euclidean Algorithm,

$40 = 5(7) + 5$ — (1)

$7 = 1(5) + 2$ — (2)

$5 = 2(2) + 1$ — (3)

We stop at the last non-zero remainder

remainder = 1

Extended Euclidean Algorithm,

$1 = 5 - 2(2)$ (From eq ③)

From eq ②,

$2 = 7 - 1(5)$

So, put this in $1 = 5 - 2(2)$

$1 = 5 - 2(7 - 1(5))$

$1 = 5 - 2(7) + 2(5)$

From eq ①,

$5 = 40 - 5(7)$

So, put this in $1 = 3(5) - 2(7)$

$1 = 3[40 - 5(7)] - 2(7)$

$1 = 3(40) - 15(7) - 2(7)$

$1 = 3(40) - 17(7)$

Compare to $1 = ?(40) - d(e)$

$\therefore$ $\boxed{d = -17}$ & $\boxed{e = 7}$

↓ negative

So, $d = d \bmod \phi(n)$

$= -17 \bmod 40 = 23$

$\boxed{d = 23}$

private Key				public	
p	q	$\phi(n)$	d	n	e
11	5	40	23	55	7

} RSA works

IMP

* Alice uses n & e to encrypt & Bob uses only his private key to decrypt Alice messages.

Security behind it:

Thus, the only thing preventing someone from decrypting a publicly encoded RSA message, is that they do not know **ϕ(n).** And, the only way to find out that number is to get it from p*q, which would require factoring p*q, and nobody knows how to do that efficiently. Therefore, the only people who can decrypt publicly encoded **RSA messages**, are the people who created **p*q** in the first place, because only they know p and q.