A small RSA example.

```
> p := 79; q := 131;
```

$$p := 79$$

$$q := 131$$

p and q are randomly chosen primes. In RSA we should take p as follows:
Take a random integer p larger than 10^100.

If p is not prime, take another random integer (repeat this until p is prime).

Do the same to get q.

```
> isprime(p);
```

true

```
> isprime(q);
```

true

```
> n := p*q;
```

$$n := 10349$$

```
> phi := (p-1)*(q-1);
```

$$\varphi := 10140$$

In RSA we should take e as follows: take some random big integer e.

If gcd(e,phi) is not 1 then take another random e (repeat until gcd is 1).

```
> e := 12343; gcd(e, phi);
```

$$e := 12343$$

$$1$$

```
> d*e + t*phi=1;
```

$$12343\, d + 10140\, t = 1$$

```
> isolve(%);
```

$$\{d = 5887 + 10140\, \_Z1,\, t = -7166 - 12343\, \_Z1\}$$

isolve = find all integer solutions of that equation (this uses the Euclidean Algorithm).

There is one free variable _Z1. We get a solution of d*e + t*phi = 1

for any integer value of _Z1.

We're only interested in one solution, so we substitute some integer (namely 0) for _Z1.

```
> subs(_Z1=0,%);
```

$$\{d = 5887,\, t = -7166\}$$

```
> d := 5887;
```

$$d := 5887$$

Now we turn text to a number as follows:

a = 01   b = 02   c = 03 ... i = 09 ... z = 26 A = 27 B = 28 ... H = 34 ... Z = 52

So the message "Hi" becomes 3409.

```
> m := 3409;
```

$$m := 3409$$

Now the public key is this:

```
> public := [n, e];
```

$$public := [10349, 12343]$$

and the private key is this:

```
> private := [n,d];
```

$$private := [10349, 5887]$$

Anyone has access to the public key, so anyone can use the public key to encrypt messages.

The message m, and the encrypted message is m^e mod n.

```
> c := m^e mod n;
```

$$c := 6062$$

This c is now transmitted over the internet. Anything that goes over the internet is easy to wiretap.

So eavesdropping is easy, however, an eavesdropper won't get the message m = 3409 = Hi

Instead, the eavesdropper sees only c = 6062.

How to decrypt the encrypted message c using the private key? Well, m = c^d mod n.

```
> c^d mod n;
```

$$3409$$

So given the encrypted message c, we can decrypt it and get the original message m = 3409 = Hi

Why can't an eavesdropper do the same, why can't an eavesdropper also compute c^d mod n and thus find m?

Well, the public information is only n and e. But d is not public, it is private information. The numbers p, q and phi should also be kept a secret.

To compute d, we used the number phi, and to compute the number phi, we needed to know p and q. For an eavesdropper to compute d, he'd have to figure out what p and q are. Now n = p*q and n is public information. So if the eavesdropper can factor n, then he can find p and q, then he can find phi = (p-1)*(q-1) and then he can find d in the same way as we did.

The security of RSA now hangs on the following: if we choose very big (more than 10^100) random prime numbers p and q, and we take n = p*q, and we make n public, then it is very difficult to factor this big number n, so an eavesdropper can not find p and q even though n=p*qis public information.