restart;

I. La methode RSA

```
inverse := proc(m, p, q)
 local d, k, N;
 N := (p − 1)·(q − 1);
 d := igcdex(m, N,'u','v');
 k := u mod N;
 k;
 end;
```

```
proc(m, p, q)                                                    (1)
    local d, k, N;
    N := (p − 1) * (q − 1); d := igcdex(m, N, 'u', 'v'); k := mod(u, N); k
end proc
```

```
fm := proc(x, m, M)
 (Power(x, m) mod M);
 end;
```

```
proc(x, m, M) mod (Power(x, m), M) end proc                      (2)
```

```
m1 := 19;
                                19                               (3)
```

```
p1 := 17;
 isprime(p1);
                                17
                              true                               (4)
```

```
q1 := 127;
 isprime(q1);
                               127
                              true                               (5)
```

```
N1 := (p1 − 1)·(q1 − 1); M1 := p1·q1; igcd(m1, N1); k1 := inverse(m1, p1, q1);
                              2016
                              2159
                                1
                               955                               (6)
```

```
fm(x, m1, M1); fm(x, k1, M1); fm(fm(58, m1, M1), k1, M1);
```

$$x^{19}$$

$$x^{955}$$

$$58 \quad \textbf{(7)}$$

$secret1 := 26$;

$$26 \quad \textbf{(8)}$$

$code := fm(secret1, m1, M1)$;

$$1732 \quad \textbf{(9)}$$

$decode := fm(code, k1, M1)$;

$$26 \quad \textbf{(10)}$$

$m2 := 2^{2^4} + 1$; $isprime(m2)$;

$$65537$$

$$true$$

$p2 := 11! + 1$; $isprime(p2)$;

$$39916801$$

$$true \quad \textbf{(12)}$$

$q2 := 6^{30} + 7$; $isprime(q2)$;

$$2210739197207333578997 83$$

$$true \quad \textbf{(13)}$$

$N2 := (p2 - 1) \cdot (q2 - 1)$; $M2 := p2 \cdot q2$; $igcd(m2, N2)$; $k2 := inverse(m2, p2, q2)$;

$$882456343870856930061401813760 0$$

$$882456365978248902134741595418 3$$

$$1$$

$$576881412644141835217062695987 3 \quad \textbf{(14)}$$

$m2 \cdot k2$ **mod** $N2$;

$$1 \quad \textbf{(15)}$$

fm(x, m2, M2); $fm(x, k2, M2)$;

$$x^{65537}$$

$$x^{576881412644141835217062695987 3} \quad \textbf{(16)}$$

$fm(fm(5, m2, M2), k2, M2)$;

$$5 \quad \textbf{(17)}$$

$secret2 := 250$;

$$250 \quad \textbf{(18)}$$

$code := fm(secret2, m2, M2)$;

$$722517361114881963899448448865 4 \quad \textbf{(19)}$$

$decode := fm(code, k2, M2)$;

$$250 \quad \textbf{(20)}$$

II Preuves sans apport d'information

```
with(Maplets[Elements]):

maplet1 := Maplet(InputDialog_ID1("Choisissez y et entrez y^2",'onapprove'= Shutdown([ID1]),
    'oncancel'= Shutdown()));
maplet2 := Maplet(InputDialog_ID1("Entrez à présent y",'onapprove'= Shutdown([ID1]),'oncancel'
    = Shutdown()));
maplet3 := Maplet(InputDialog_ID1("Entrez à présent x·y",'onapprove'= Shutdown([ID1]),'oncancel'
    = Shutdown()));


roll := rand(0..1):

verif :=proc(a, p, q)
 local yy, y, cc, c, bb, b, d, C, Y, A, N;
 N := (p − 1)·(q − 1);
 bb := Maplets[Display](maplet1);
 b := parse(bb[1]);
 d := roll();
 if d = 1 then yy := Maplets[Display](maplet2);
               y := parse(yy[1]);
               Y := y·y mod N;
               if Y ≠ b then return 0;
                         else return 1;
               fi;
          else cc := Maplets[Display](maplet3);
               c := parse(cc[1]);
               C := c·c mod N; A := a·b mod N;
               if C ≠ A then  return 0;
                         else return 1;
               fi;
 fi;
 end;
```

```
proc(a, p, q)                                                    (21)
    local yy, y, cc, c, bb, b, d, C, Y, A, N;
    N := (p − 1) * (q − 1);
```

```
    bb := Maplets[Display](maplet1);
    b := parse(bb[1]);
    d := roll( );
    if d = 1 then
        yy := Maplets[Display](maplet2);
        y := parse(yy[1]);
        Y := mod(y * y, N);
        if Y <> b then return 0 else return 1 end if
    else
        cc := Maplets[Display](maplet3);
        c := parse(cc[1]);
        C := mod(c * c, N);
        A := mod(a * b, N);
        if C <> A then return 0 else return 1 end if
    end if
end proc

verif(16, 17, 11);
```

$$0 \tag{22}$$

```
boucle := proc(a, p, q, n)
 local i, G;
 i := 0;
 G := 1;
 while (i < n and G = 1) do G := verif(a, p, q);
                                i := i + 1;
                            od ;
 if G = 1 then print(Tu connais peut etre); print(le secret);
        else print(Menteur tu ne connais); print( pas le secret);
 fi;
 end;
proc(a, p, q, n)
    local i, G;
    i := 0;
    G := 1;
    while i < n and G = 1 do G := verif(a, p, q); i := i + 1 end do;
    if G = 1 then
        print(Tu* connais* peut* etre); print(le* secret)
    else
        print(Menteur* tu* ne* connais); print(pas* le* secret)
    end if
```

end proc

boucle(16, 11, 17, 5);

Tu connais peut etre

le secret **(24)**

Power(8, -1) **mod** 55;

```
Error, (in mod/Power) 2nd argument (exponent) must be a non-
negative integer
```

8&^(-1) **mod** 55;

```
Error, invalid base
```

8&^(-1) **mod** 55;

?*Inverse*