```
┌──────────────────────── MODULE rsa ────────────────────────┐
EXTENDS Integers, Sequences, FiniteSets

VARIABLES p, q, n, phi, e, d, m, c, message, ciphertext, plaintext

Prime ≜ {x ∈ 60 . . 100 : ∀ y ∈ 2 . . (x − 1) : x%y ≠ 0}

ChoosePrime ≜ CHOOSE x ∈ Prime : TRUE

RECURSIVE ModExp(_, _, _)
ModExp(base, exp, mod) ≜
  IF exp = 0 THEN 1
   ELSE
    IF exp%2 = 0 THEN
      LET half_exp ≜ ModExp(base, exp ÷ 2, mod)IN
      (half_exp ∗ half_exp)%mod
     ELSE
      (base ∗ ModExp(base, exp − 1, mod))%mod

RECURSIVE ExtendedGCD(_, _, _, _, _, _)
ExtendedGCD(a, b, x0, y0, x1, y1) ≜
  IF b = 0 THEN ⟨a, x0, y0⟩
   ELSE
    LET q_ ≜ a ÷ b
        r_ ≜ a%b
    IN  ExtendedGCD(b, r_, x1, y1, x0 − q_ ∗ x1, y0 − q_ ∗ y1)

InverseMod(a, m_) ≜
  LET gcdResult ≜ ExtendedGCD(a, m_, 1, 0, 0, 1)
      gcd ≜ gcdResult[1]
      x_ ≜ gcdResult[2]
  IN  IF gcd = 1 THEN (x_ + m_)%m_ ELSE 0

GenerateKeys ≜
    ∧ d′ = InverseMod(e, phi)
    ∧ UNCHANGED ⟨p, q, n, phi, e, m, c, plaintext, ciphertext, message⟩


Encrypt ≜
    ∧ c′ = ModExp(m, e, n)
    ∧ UNCHANGED ⟨p, q, n, phi, e, d, m, plaintext, ciphertext, message⟩

Decrypt ≜
    ∧ plaintext′ = ModExp(c, d, n)
    ∧ UNCHANGED ⟨p, q, n, phi, e, d, m, c, ciphertext, message⟩

Output ≜
    ∧  ciphertext′ = c
    $\land\ message' = plaintext$
    $\land$ UNCHANGED $\langle p,\ q,\ n,\ phi,\ e,\ d,\ m,\ c,\ plaintext \rangle$

$Next\ \triangleq$
    $\lor\ GenerateKeys$
    $\lor\ Encrypt$
    $\lor\ Decrypt$
    $\lor\ Output$

$Init\ \triangleq$
    $\land\ p = ChoosePrime$
    $\land\ q =$ CHOOSE $x \in Prime : x \neq p$
    $\land\ n = p * q$
    $\land\ phi = (p - 1) * (q - 1)$
    $\land\ e = 65537$
    $\land\ ExtendedGCD(e,\ phi,\ 1,\ 0,\ 0,\ 1)[1] = 1$
    $\land\ d\ = InverseMod(e,\ phi)$
    $\land\ m \in 1\ ..\ (n - 1)$
    $\land\ c\ = ModExp(m,\ e,\ n)$
    $\land\ plaintext = ModExp(c,\ d,\ n)$
    $\land\ ciphertext = c$
    $\land\ message = plaintext$

$Spec\ \triangleq$
  $Init \land \Box[Next]_{\langle p,\, q,\, n,\, phi,\, e,\, d,\, m,\, c,\, plaintext,\, ciphertext,\, message \rangle}$
```