All of the scripts that were produced for Maxima in the Cryptography Notes Technology Guide are contained in the CryptDS.mac file. To load this into a Maxima session,

1. Select File > Load Package from the main menu.
2. Navigate to the CryptDS.mac file.
3. Select it and click Open.

At this point Maxima will invoke the load command and all of the scripts will be loaded.

Script List:

```
from_cf(L)             ---  Converts the list L into a continued fraction and simplifies.
from_cf_n(L, n)        ---  Converts the list L into a continued fraction using the first
                            n entries of the list and simplifies.

rref(A)                ---  Returns the reduced row echelon form of the matrix A.
mat_mod_inverse(M, n)  ---  Finds the inverse of the square matrix M modulo n, if it exists.
mod_echelon(a,n)       ---  Reduces the Matrix A to an echelon form modulo n.  If echelon
                            form is not possible it will take the reduction as far as it can.
mod_rref(a,n)          ---  Reduces the Matrix A to reduced row echelon form modulo n.  If
                            reduced row echelon form is not possible it will take the
                            reduction as far as it can.

ec_points(b,c,n)       ---  Calculates all points on the Elliptic Curve y^2 = x^3+bx+c (mod n).
                            Points are returned as a list of ordered pair lists. The point at
                            infinity is not included in this list.
ec_allpoints(b,c,n)    ---  Calculates all points on the Elliptic Curve y^2 = x^3+bx+c (mod n).
                            Points are returned as a list of ordered pair lists. The point at
                            infinity is included in this list.
ec_order(b,c,n)        ---  Calculates the number of points on the Elliptic Curve y^2 = x^3+bx+c (mod n).
                            The point at infinity is included in this count.
ec_pointAdd(b,c,n,p1,p2) ---  Calculates the sum of p1 and p2 on the Elliptic Curve y^2 = x^3+bx+c (mod n).
                            The returned point is an ordered pair list. The points P1 and P2 must be
                            in this form as well.  For example, [1,2] and [7,3].
ec_pointScalarMult(b,c,n,t,p1) --- Calculates the scalar multiple t * p1 on the Elliptic Curve
                            y^2 = x^3+bx+c (mod n). The returned point is an ordered pair list. The
                            point p1 must be in this form as well.  For example, [7,3].
ec_pointFactorialScalarMult(b,c,n,t,p1) --- Calculates the scalar factorial multiple t! * p1 on the
                            Elliptic Curve y^2 = x^3+bx+c (mod n). The returned point is an ordered pair list. The
                            point p1 must be in this form as well.  For example, [7,3].
ec_pointOnCurve(b,c,n,pt) --- Returns true if the point pt is on the Elliptic Curve y^2 = x^3+bx+c (mod n)
                            and false if it is not. The point pt must be in list form, for example, [7,3].
ec_pointWithX(b,c,n,x) ---  Finds a point on the Elliptic Curve y^2 = x^3+bx+c (mod n) with the given
                            x coordinate.  If no point is found the function returns "none".
ec_pointWithY(b,c,n,y) ---  Finds a point on the Elliptic Curve y^2 = x^3+bx+c (mod n) with the given
                            y coordinate.  If no point is found the function returns "none".
ec_generateCurveConstant(b,n,pt) --- Returns the needed constant term c so that the point pt is on the
                            Elliptic Curve y^2 = x^3+bx+c (mod n). The point pt must be in list form, for example, [7,3].
ec_pointOrder(b,c,n,pt) --- Finds the order of the point on the Elliptic Curve y^2 = x^3+bx+c (mod n).
                            The point pt must be in list form, for example, [7,3].
ec_plot(b,c,n)         ---  Calculates all of the finite points on the Elliptic Curve y^2 = x^3+bx+c (mod n)
                            and then plots the points.
```