The ECP-128 Library version 1.1

Email: witeg@mail.com
Homepage: http://www.witeg.prv.pl

Introduction:

The ECP-128 Library provides an elliptic curve arithmetic in GF(p). The ECP-128 implements operations on elliptic curve verifiably at random – *RandomCurve1-P128-WiteG*.

```
RandomCurve1-P128-WiteG:
p       340282366920938463444927863358058659863
seedE   0x9E39F75ADE0AE5CFDBE0BD847F7B7EAFC484C48F
r       0x5E0AE5CFDBE0BD847F7B7EAFC484C48F
a       -3
b       103744651967215942079424252318256895516
xG      0x504E0BD39A2B41A161174BA8FD79309F
yG      0x9A45FA6D7279A790BB0D8845D469DED4
n       340282366920938463450938462077435853809
h       1
```

Exported functions:

The **ECP_A2J** function converts point coordinates from affine to Jacobian representation:

VOID ECP_A2J(
    [IN]   **BYTE**   **pbPointAffine*
    [OUT]   **BYTE**   **pbPointJacobian*
);

Parameters:
pbPointAffine
    The address of point in affine coordinates.
pbPointJacobian
    The address of the buffer to receive point in Jacobian coordinates.

Return Value:
This function does not return a value.

The **ECP_Add** function adds two affine points on the elliptic curve:

VOID ECP_Add(
    [IN]   **BYTE**   **pbPointAffineA*
    [IN/OUT]   **BYTE**   **pbPointAffineB*
);

Parameters:
pbPointAffineA
    The address of point A in affine coordinates.

pbPointAffineB
The address of point B in affine coordinates. On exit *B=B+A* (in affine coordinates).

Return Value:
This function does not return a value.

The **ECP_Add_J** function adds two Jacobian points:

VOID ECP_Add_J(
[IN] **BYTE** **pbPointJacobianA*
[IN/OUT] **BYTE** **pbPointJacobianB*
);

Parameters:
pbPointAffineA
The address of point A in Jacobian coordinates.
pbPointAffineB
The address of point B in Jacobian coordinates. On exit *B=B+A* (in Jacobian coordinates).

Return Value:
This function does not return a value.

The **ECP_Copy** function copies one affine point to another:

VOID ECP_Copy(
[IN] **BYTE** **pbPointAffineA*
[OUT] **BYTE** **pbPointAffineB*
);

Parameters:
pbPointAffineA
The address of affine point to be copied.
pbPointAffineB
The address of the buffer to receive the point *pbPointAffineA.*

Return Value:
This function does not return a value.

The **ECP_Dbl** function implements an elliptic curve point doubling using affine coordinates:

VOID ECP_Dbl(
[IN] **BYTE** **pbPointAffineA*
[OUT] **BYTE** **pbPointAffineB*
);

Parameters:
pbPointAffineA
The address of point A in affine coordinates.
pbPointAffineB
The address of the buffer to receive the point *B=2*A* (in affine coordinates).

Return Value:
This function does not return a value.

The **ECP_Dbl_J** function implements an elliptic curve point doubling using Jacobian coordinates:

```
VOID ECP_Dbl_J(
        [IN]            BYTE            *pbPointJacobianA
        [OUT]           BYTE            *pbPointJacobianB
);
```

Parameters:

pbPointAffineA
The address of point A in Jacobian coordinates.

pbPointAffineB
The address of the buffer to receive the point $B=2*A$ (in Jacobian coordinates).

Return Value:

This function does not return a value.

The **ECP_J2A** function converts point coordinates from Jacobian to affine representation:

```
VOID ECP_J2A(
        [IN]            BYTE            *pbPointJacobian
        [OUT]           BYTE            *pbPointAffine
);
```

Parameters:

pbPointAffine
The address of point in Jacobian coordinates.

pbPointJacobian
The address of the buffer to receive point in affine coordinates.

Return Value:

This function does not return a value.

The **ECP_Mul** function multiplies an affine point on the elliptic curve by an integer:

```
VOID ECP_Mul(
        [IN]            BYTE            *pbIntK
        [IN]            BYTE            *pbPointAffineA
        [OUT]           BYTE            *pbPointAffineB
);
```

Parameters:

pbIntK
The address of integer k.

pbPointAffineA
The address of point A in affine coordinates.

pbPointAffineB
The address of the buffer to receive the affine point $B=k*A$.

Return Value:

This function does not return a value.

The **ECP_Zero** function clears an affine point:

VOID ECP_Zero(
[OUT] **BYTE** **pbPointAffine*
);

Parameters:
pbPointAffine
The address of point in affine coordinates

Return Value:
This function does not return a value.

The **ECP_Zero_J** function clears a Jacobian point:

VOID ECP_Zero_J(
[OUT] **BYTE** **pbPointJacobian*
);

Parameters:
pbPointJacobian
The address of point in Jacobian coordinates

Return Value:
This function does not return a value.

The **set_N** function sets the elliptic curve order (*n*) as a modulus for modular arithmetic:

VOID set_N(void);

Parameters:
This function has no parameters.

Return Value:
This function does not return a value.

The **set_P** function sets the size of the elliptic curve underlying field (*p*) as a modulus for modular arithmetic:

VOID set_P(void);

Parameters:
This function has no parameters.

Return Value:
This function does not return a value.

History version :

14.05.2006 - version 1.0
20.05.2006 - version 1.1, bugfix in ECP_Zero_J

License :

"Software" means the program supplied by WiteG herewith.

Permission is hereby granted to any individual, organization or agency to use the Software for any legal NON-COMMERCIAL purpose, without any obligation to the author. You may distribute the Software freely, provided that the original distribution package (binaries and any other files included in it) is left intact. You may also disassemble, reverse engineer or modify the Software, but you MAY NOT distribute it in modified form.
Any COMMERCIAL use of the Software without commercial license obtained from the author is strictly prohibited.

Warranty:

This software is provided by WiteG as-is, without warranty of ANY KIND, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose. The author shall NOT be held liable for ANY damage to you, your computer, or to anyone or anything else, that may result from its use, or misuse. Use it at YOUR OWN RISK.
However, the author of this software does hereby declare that there are no hidden weaknesses or trapdoors inserted by him.

Please report any bugs, comments, questions or suggestions to me