Algorithms Of Representation

Generic Group $G_n^{a,b}$

Chillali Abdelhakim

FST DE FEZ
Chil2015@yahoo.fr

Abstract. In this work we study the algorithms properties for the group $E_n^{a,b}$. We establish that $G_n^{a,b}$ is a generic group. The implementations from the algorithms are by Maple.

Keywords: Elliptic curves; Cryptography; Generic group; Finite field.

1 Introduction:

Groups where the discrete logarithm problem (DLP) is believed to be intractable have proved to be inestimable building blocks for cryptographic applications. They are at the heart of numerous protocols such as key agreements, public-key cryptosystems, digital signatures, identification schemes, publicly verifiable secret sharings, hash functions and bit commitments. The search for new groups with intractable DLP is therefore of great importance. We study the algorithms properties for the group $E_n^{a,b}$.[1, 2, 3, 4, 5]

Let p be an odd prime number and n be an integer such that $n \geq 1$. Consider the quotient ring $A_n = {\mathbb{F}_q[X]}/{(X^n)}$ where $\mathbb{F}_q$ is the finite field of order q and characteristic p.

Then the ring A_n may be identified to the ring $\mathbb{F}_q[\varepsilon]$ where $\boldsymbol{\varepsilon^n = 0}$. In other word

$$A_n = \left\{ \sum_{i=0}^{n-1} a_i \, \varepsilon^i | (a_i)_{0 \leq i \leq n-1} \in \mathbb{F}_q^n \right\}.$$

Notation 1.1 We denote the canonical projections

$$\pi_k : \left| \begin{array}{c} A_k \longrightarrow A_{k-1} \\ \sum_{i=0}^{k-1} a_i \varepsilon^i \mapsto \sum_{i=0}^{k-2} a_i \rho^i \end{array} \right.$$

$$k^{\pi}: \left| \begin{array}{c} A_k \longrightarrow A_1 \\ \sum_{i=0}^{k-1} a_i \varepsilon^i \mapsto a_0 \end{array} \right.$$

An elliptic curve over a ring A_n is a curve that is given by an equation of the form

$$Y^2Z = X^3 + aXZ^2 + bZ^3, \qquad (*)$$

where $a, b \in A_n$ and $4a^3 + 27b^2$ is invertible in A_n.

We denote by $E_n^{a,b}$ the elliptic curve over A_n.

The set $E_n^{a,b}$ is an abelian group. Its identity element is a special point $\mathcal{O}$called the point at infinity.

2 Representation of elements from A_n:

Let $X = \sum_{i=0}^{n-1} x_i \, \varepsilon^i$ in A_n then, we choose to represent X by $\bar{X} = [x_0, x_1, \dots, x_{n-1}]$.

2.1 Algorithm(1)

The algorithm(1) is an algorithm that calculates the sum of two elements from A_n.

Table 1. Algorithm(1)

1. Somme(p)

- Inputs: $\bar{X}$, $\bar{Y}$
- Output: $\bar{X} + \bar{Y}$

(a) return($[x_0 + y_0, \dots, x_{n-1} + y_{n-1}]$ mod p)

2. The procedure sommep:

- sommep:=proc(X, Y,p)

return($X + Y$ mod p);

end:

2.2 Algorithm(2)

The algorithm(2) is an algorithm that calculates the product of two elements from A_n.

Table 2. Algorithm(2)

1. Produit(p)

- Inputs: $\bar{X}, \bar{Y}$
- Output: $\bar{X}\bar{Y}$

(a) $z_0 := x_0 y_0 \ mod \ p$

(b)

— for j from 1 to n-1 do
— for i from 1 to j do
— $z_j := z_0 + \sum_{i=1}^{j} x_i y_{j-i} \ mod \ p$
— end for
— end for

(c) return($[z_0, z_1, \dots, z_{n-1}]$)

2. The procedure produitp:

- produitp:=proc(X, Y,p)

```
Local n, i, j, x;
n:= nops(X);
for j from 1 to n do
 for i from 1 to j do
          x[j]:=add(X[i]*Y[j-i+1], i=1..j mod p);
 end do;
end do;
retourn([seq(x[i], i=1..n]);
end:
```

2.3 Algorithm(3)

The algorithm(3) is an algorithm that calculates the inverse of an element from A_n.

Table 3. Algorithm(3)

1. Inverse(p)

- Input: $\bar{X}$
- Output: inverse

(a) if $x_0 := 0$ then print(Not invertible)

(b) else

(c) $y_0 := x_0^{-1} \ mod \ p$;

(d) for i from 0 to n-1 do for j from 1 to i do $y_j := -y_0 \sum_{i=0}^{j-1} y_i x_{j-i} \ mod \ p$

(e) end for

(f) end for

(g) return($[y_0, y_1, \dots, y_{n-1}]$)

(h) end if

2. The procedure inversep:

- inversep:=proc(X,p)
  local i, j, x, n;
  n:=nops(X);
  if (X[1]=0) then print("Not invertible");
  else x[1]:=expand((X[1])^-1 mod p);
  for i from 1 to n do
  for j from 2 to i do
  x[j]:=expand((-x[1]*add(x[i]*X[j-i+1], i=1..j-1) mod p);
  end do;
  end do;
  return([seq(x[i],i=1..n)]);
  end if;

end:

2.4 Algorithm(4)

The algorithm(4) is an algorithm which gives the projection of an element from A_n to A_1.

Table 4. Algorithm(4)

1. Projection(1)

- Input: $\bar{X}$
- Output: x_0

(a) return(x_0)

2. The procedure projection1:

- projection1:=proc(X)
  return(X[1]);

end:

2.5 Algorithm(5)

The algorithm(5) is an algorithm which gives the projection of an element from A_n to A_{n-1}.

Table 5. Algorithm(5)

1. Projection(2)

- Input: $\bar{X}$
- Output: $[x_0, x_1, \dots, x_{n-2}]$

(a) return($[x_0, x_1, \dots, x_{n-2}]$)

2. The procedure projection2:

- projection2:=proc(X)
  local n;
  n:=nops(X);
  return([seq(X(i),i=1..n-1]);

end:

Remark2.2 This algorithms define on A_n have a running time:

- For inverse i(p,n).
- For sum s(p,n).
- For multiplication m(p,n).

The execution time depends on n and p. We have:

$$s(p,n) \leq m(p,n) \leq i(p,n) \leq clog(p).$$

This algorithms are a polynomial time algorithms.

3 The generic group $G_n^{a,b}$

Let $[X:Y:Z] \in E_n^{a,b}$. Then we represent $[X:Y:Z]$ by:

$$[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}].$$

Notation 3.3 We denote the representative algorithms of $E_n^{a,b}$, the set $G_n^{a,b}$, it's formed by $[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}]$, where x_i, y_i and z_i are elements of $\mathbb{F}_p$, for i from 0 to n-1.

3.1 Algorithm(6)

This algorithm is used to represent the identity element of $E_n^{a,b}$.

Table 6. Algorithm(6)

1. Neutre

- Input: n
- Output: $[\underbrace{0, \dots, 0}_{n}, 1, \underbrace{0,0, \dots, 0}_{2n-1}]$

(a) $[\underbrace{0, \dots, 0}_{n}, 1, \underbrace{0,0, \dots, 0}_{2n-1}]$

2. The procedure Neut:

- Neut:=proc(n)
  return([seq(0,i=1…n),1,seq(0,i=n+2..3*n)];
end:

3.2 Algorithm(7)

This is an algorithm that represents the projection π^n.

Table 7. Algorithm(7)

1. Projection(a)

- Input: $[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}]$
- Output: $[x_0, x_1, \dots, x_{n-2}, y_0, y_1, \dots, y_{n-2}, z_0, z_1, \dots, z_{n-2}]$
  (a) retourner $[x_0, x_1, \dots, x_{n-2}, y_0, y_1, \dots, y_{n-2}, z_0, z_1, \dots, z_{n-2}]$

2. The procedure proj

- proj:=proc(P)
  local S,T,R,n,L,M,N,i;
  n:=nops(P)/3;
  S:=[seq(P[i],i=1..n)];
  T:=[seq(P[i],i=n+1..2*n)];
  R:=[seq(P[i],i=2*n+1..3*n)];
  L:=projection2(S);
  M:=projection2(T);
  N:=projection2(R);
  return([seq(L[i],i=1..n-1),seq(M[i],i=1..n-1),seq(N[i],i=1..n-1)]);
end:

3.3 Algorithm(8)

This is an algorithm that represents the projection n_π.

Table 8. Algorithm(8)

1. Projection(b)

- Input: $[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}]$
- Output: $[x_0, y_0, z_0\ ]$
  (a) retourner $[x_0, y_0, z_0\ ]$

2. The procedure projection

- projection:=proc(P)
  local S,T,R,n,i;
  n:=nops(P)/3;
  S:=[seq(P[i],i=1..n)];
  T:=[seq(P[i],i=n+1..2*n)];
  R:=[seq(P[i],i=2*n+1..3*n)];
  return([projection1(S), projection1(T), projection1(R)]);
end:

3.4 Algorithm(9)

This is an algorithm that verifies that an element belongs or no to $G_n^{a,b}$.

Table 9. Algorithm(9)

1. appartient

- Input: $[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}]$
- Outputs: 0 or 1
  (a) X=$[x_0, x_1, \dots, x_{n-1}]$
  (b) $Y = [y_0, y_1, \dots, y_{n-1}]$
  (c) $Z = [z_1, \dots, z_{n-1}]$
  (d) $r = produitp(produitp(Y, Y, p), Z, p)$
  (e) $s = produitp(produitp(X, X, p), X, p)$
  (f) $t = produitp(produitp(a, X, p), produitp(Z, Z, p), p)$
  (g) $l = produitp(produitp(b, Z, p), produitp(Z, Z, p), p)$
  (h) $if\ (r = sommep(s, sommep(t, l, p), p))$ then return 1
  (i) $else$ return 0
  (j) $end\ if$

2. The procedure app

- app:=proc(P,a,b,p)
  local S, Q, R, X, Y, Z, T, n;
  n:=nops(P)/3;
  S:=[seq(P[i],i=1..n)];
  T:=[seq(P[i],i=n+1..2*n)];
  R:=[seq(P[i],i=2*n+1..3*n)];
  X:= $produitp(produitp(T, T, p), R, p)$;
  Y:= $produitp(produitp(S, S, p), S, p)$;
  Z:= $produitp(produitp(a, S, p), produitp(R, R, p), p)$;
  T:= $produitp(produitp(b, R, p), produitp(R, R, p), p)$;
  if (X= $sommep(Y, sommep(Z, T, p), p)$) then return(1)
  else return(0) end if;
end:

3.5 Algorthim(10)

This is an algorithm that checks the equality of two elements of the group $G_n^{a,b}$.

Table 10. Algorithm(10)

1. Egal

- Inputs: $[x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}, z_0, z_1, \dots, z_{n-1}]$, $[s_0, s_1, \dots, s_{n-1}, t_0, t_1, \dots, t_{n-1}, l_0, l_1, \dots, l_{n-1}]$
- Outputs: 0 or 1

(a) sortie:=0
(b) if ($z_0 = 0$ and $l_0 = 0$) then

— $r = produitp([x_0, x_1, \dots, x_{n-1}], inversep([y_0, y_1, \dots, y_{n-1}], p, p)$
— s = $produitp([s_0, s_1, \dots, s_{n-1}], inversep([t_0, t_1, \dots, t_{n-1}], p, p)$
— l = $produitp([z_0, z_1, \dots, z_{n-1}], inversep([y_0, y_1, \dots, y_{n-1}], p, p)$
— t = $produitp([l_0, l_1, \dots, l_{n-1}], inversep([t_0, t_1, \dots, t_{n-1}], p, p)$

(c) if (r=s and l=t) then sortie:=1
(d) end if
(e) else

— $r = produitp([x_0, x_1, \dots, x_{n-1}], inversep([z_0, z_1, \dots, z_{n-1}], p, p)$
— s = $produitp([s_0, s_1, \dots, s_{n-1}], inversep([l_0, l_1, \dots, l_{n-1}], p, p)$
— l = $produitp([z_0, z_1, \dots, z_{n-1}], inversep([y_0, y_1, \dots, y_{n-1}], p, p)$
— t = $produitp([l_0, l_1, \dots, l_{n-1}], inversep([t_0, t_1, \dots, t_{n-1}], p, p)$

(f) if (r=s and l=t) then sortie:=1
(g) end if
(h) else
(i) return (sortie)

2. The procedure Egal1

- Egal1:=proc(P,Q,a,b,p)

```
local S,sortie,T,R,E,F,G,X,Y,Z,n,i;
sortie:=0;
n:=nops(P)/3;
S:=[seq(P[i],i=1..n)];
T:=[seq(P[i],i=n+1..2*n)];
R:=[seq(P[i],i=2*n+1..3*n)];
E:=[seq(Q[i],i=1..n)];
F:=[seq(Q[i],i=n+1..2*n)];
G:=[seq(Q[i],i=2*n+1..3*n)];
X:=produitp(S,inversep(T,p),p)-produitp(E,inversep(F,p),p) mod p;
Y:=produitp(R,inversep(T,p),p)-produitp(G,inversep(F,p),p) mod p;
```

```
Z:=[seq(X[i],i=1..n),seq(Y[i],i=1..n)];
if ( Z=[seq(0,i=1..2*n)]) then sortie:=1;
end if;
return(sortie)
end :
```

3. The procedure Egal2

- Egal2:=proc(P,Q,a,b,p)

```
local S,sortie,T,R,E,F,G,X,Y,Z,n,i;
sortie:=0;
n:=nops(P)/3;
S:=[seq(P[i],i=1..n)];
T:=[seq(P[i],i=n+1..2*n)];
R:=[seq(P[i],i=2*n+1..3*n)];
E:=[seq(Q[i],i=1..n)];
F:=[seq(Q[i],i=n+1..2*n)];
G:=[seq(Q[i],i=2*n+1..3*n)];
X:=produitp(S,inversep(R,p),p)-produitp(E,inversep(G,p),p) mod p;
Y:=produitp(T,inversep(R,p),p)-produitp(F,inversep(G,p),p) mod p;
Z:=[seq(X[i],i=1..n),seq(Y[i],i=1..n)];
if ( Z=[seq(0,i=1..2*n)]) then sortie:=1;
end if;
return(sortie)
end :
```

4. The procedure Egal3

- Egal3:=proc(P,Q,a,b,p)

```
  local S,T,R,E,F,G,n,i;
  n:=nops(P)/3;
  S:=[seq(P[i],i=1..n)];
  T:=[seq(P[i],i=n+1..2*n)];
  R:=[seq(P[i],i=2*n+1..3*n)];
  E:=[seq(Q[i],i=1..n)];
  F:=[seq(Q[i],i=n+1..2*n)];
  G:=[seq(Q[i],i=2*n+1..3*n)];
  if (R[1]*G[1]=0 mod p) then  return(0) end if;
end :
```

5. The procedure Egal4

- Egal4:=proc(P,Q,a,b,p)

```
  local S,T,R,E,F,G,n,i;
  n:=nops(P)/3;
  S:=[seq(P[i],i=1..n)];
  T:=[seq(P[i],i=n+1..2*n)];
```

```
    R:=[seq(P[i],i=2*n+1..3*n)];
    E:=[seq(Q[i],i=1..n)];
    F:=[seq(Q[i],i=n+1..2*n)];
    G:=[seq(Q[i],i=2*n+1..3*n)];
    if (R[1]*G[1]=0 mod p) then Egal3(P,Q,R,E,F,G,a,b,p) else Egal2(P,Q,R,E,F,G,a,b,p) end if;
end:
```

6. The procedure Egal

```
• Egal:=proc(P,Q,a,b,p)
    local S,T,R,E,F,G,n,i;
    n:=nops(P)/3;
    S:=[seq(P[i],i=1..n)];
    T:=[seq(P[i],i=n+1..2*n)];
    R:=[seq(P[i],i=2*n+1..3*n)];
    E:=[seq(Q[i],i=1..n)];
    F:=[seq(Q[i],i=n+1..2*n)];
    G:=[seq(Q[i],i=2*n+1..3*n)];
    if (T[1]*F[1]=0 mod p) then Egal4(P,Q,a,b,p)else Egal1(P,Q,a,b,p) end if;
end:
```

3.6 The procedures p_1, p_2, p_3, p_4, p_5 and p_6

Table 11. The procedures p_1, p_2, p_3, p_4, p_5 and p_6

1. The procedure p_1

```
• p1:=proc(X,Y,a,b,p)
 local P,Q,R,E,F,G,x,y,z,t,n;
    n:=nops(X)/3;
    P:=[seq(X[i],i=1..n)];
    Q:=[seq(X[i],i=n+1..2*n)];
    R:=[seq(X[i],i=2*n+1..3*n)];
    E:=[seq(Y[i],i=1..n)];
    F:=[seq(Y[i],i=n+1..2*n)];
    G:=[seq(Y[i],i=2*n+1..3*n)];
    x:=produitp(produitp(Q,Q,p),produitp(E,G,p),p) mod p;
    y:=-produitp(produitp(F,F,p),produitp(P,R,p),p) mod p ;
    z:=-produitp(a,produitp(sommep(produitp(R,E,p),produitp(P,G,p),p),sommep(produitp(R,E,p),
   -produitp(P,G,p),p),p),p) mod p ;
    t:=produitp(sommep(2*produitp(Q,F,p),-3*produitp(produitp(b,R,p),G,p),p),
   sommep(produitp(R,E,p),-produitp(P,G,p),p),p) mod p;
    sommep(sommep(x,y,p),sommep(z,t,p),p);
```

```
end:
```

2. The procedure p_2

- p2:=proc(X,Y,a,b,p)

```
 local P,Q,R,E,F,G,x,y,z,t,n;
 n:=nops(X)/3;
 P:=[seq(X[i],i=1..n)];
 Q:=[seq(X[i],i=n+1..2*n)];
 R:=[seq(X[i],i=2*n+1..3*n)];
 E:=[seq(Y[i],i=1..n)];
 F:=[seq(Y[i],i=n+1..2*n)];
 G:=[seq(Y[i],i=2*n+1..3*n)];
 x:=produitp(produitp(Q,F,p),sommep(produitp(G,Q,p),-produitp(R,F,p),p),p) mod p;
 y:=-produitp(a,sommep(produitp(produitp(Q,P,p),produitp(G,G,p),p),-produitp
(produitp(E,F,p),produitp(R,R,p),p),p),p) mod p;
 z:=produitp(sommep(-2*produitp(a,produitp(R,G,p),p),-3*produitp(P,E,p),p),
sommep(produitp(E,Q,p),-produitp(P,F,p),p),p) mod p ;
 t:=-3*produitp(produitp(b,produitp(G,R,p),p),sommep(produitp(G,Q,p),
-produitp(R,F,p),p),p) mod p;
 sommep(sommep(x,y,p),sommep(z,t,p),p);
end:
```

3. The procedure p_3

- p3:=proc(X,Y,a,b,p)

```
 local P,Q,R,E,F,G,x,y,z,t,n;
n:=nops(X)/3;
P:=[seq(X[i],i=1..n)];
Q:=[seq(X[i],i=n+1..2*n)];
R:=[seq(X[i],i=2*n+1..3*n)];
E:=[seq(Y[i],i=1..n)];
F:=[seq(Y[i],i=n+1..2*n)];
G:=[seq(Y[i],i=2*n+1..3*n)];
x:=produitp(sommep(produitp(R,F,p),produitp(G,Q,p),p),sommep(produitp(G,Q,p),
-produitp(R,F,p),p),p) mod p;
y:=produitp(sommep(3*produitp(P,E,p),produitp(produitp(G,a,p),R,p),p),sommep
(produitp(E,R,p),-produitp(P,G,p),p),p) mod p;
sommep(x,y,p);
end:
```

4. The procedure p_4

- p4:=proc(X,Y,a,b,p)

```
local P,Q,R,E,F,G,x,y,z,t,n;
```

```
    n:=nops(X)/3;
    P:=[seq(X[i],i=1..n)];
    Q:=[seq(X[i],i=n+1..2*n)];
    R:=[seq(X[i],i=2*n+1..3*n)];
    E:=[seq(Y[i],i=1..n)];
    F:=[seq(Y[i],i=n+1..2*n)];
    G:=[seq(Y[i],i=2*n+1..3*n)];
    x:=produitp(sommep(produitp(F,Q,p),-6*produitp(produitp(R,b,p),G,p),p),
    sommep(produitp(E,Q,p),produitp(P,F,p),p),p) mod p;
    y:=produitp(sommep(produitp(produitp(a,a,p),produitp(R,G,p),p),
    -2*produitp(produitp(a,P,p),E,p),p),sommep(produitp(R,F,p),produitp(G,Q,p),p),p) mod p;
    z:=-produitp(3*b,sommep(produitp(produitp(P,Q,p),produitp(G,G,p),p),produitp
    (produitp(R,R,p),produitp(F,E,p),p),p),p) mod p ;
    t:=-produitp(a,sommep(produitp(produitp(R,Q,p),produitp(E,E,p),p),produitp
    (produitp(P,P,p),produitp(F,G,p),p),p),p) mod p ;sommep(sommep(x,y,p),sommep(z,t,p),p);
end:
```

5. The procedure p_5

- p5:=proc(X,Y,a,b,p)

```
    local P,Q,R,E,F,G,x,y,z,t,n;
    n:=nops(X)/3;
    P:=[seq(X[i],i=1..n)];
    Q:=[seq(X[i],i=n+1..2*n)];
    R:=[seq(X[i],i=2*n+1..3*n)];
    E:=[seq(Y[i],i=1..n)];
    F:=[seq(Y[i],i=n+1..2*n)];
    G:=[seq(Y[i],i=2*n+1..3*n)];
    x:=sommep(sommep(produitp(produitp(Q,Q,p),produitp(F,F,p),p),produitp(3*produitp(a,
  produitp(P,P,p),p),produitp(E,E,p),p),p),produitp(sommep(-produitp(a,produitp(a,a,p),p),
  -9*produitp(b,b,p),p),produitp(produitp(R,R,p),produitp(G,G,p),p),p),p) mod p;
    y:=produitp(-produitp(a,a,p),produitp(sommep(produitp(R,E,p),produitp(P,G,p),p),sommep
  (produitp(R,E,p),produitp(P,G,p),p),p),p) mod p;
    z:=produitp(-2*produitp(produitp(a,a,p),produitp(produitp(R,P,p),E,p),p),G,p) mod p ;
    t:=produitp(sommep(9*produitp(produitp(b,P,p),E,p),-3*produitp(produitp(a,b,p),
  produitp(R,G,p),p),p),sommep(produitp(R,E,p),produitp(P,G,p),p),p) mod p ;
    sommep(sommep(x,y,p),sommep(z,t,p),p);
 end:
```

6. The procedure p_6

- p6:=proc(X,Y,a,b,p)

```
    local P,Q,R,E,F,G,x,y,z,t,n;
    n:=nops(X)/3;
    P:=[seq(X[i],i=1..n)];
```

```
    Q:=[seq(X[i],i=n+1..2*n)];
    R:=[seq(X[i],i=2*n+1..3*n)];
    E:=[seq(Y[i],i=1..n)];
    F:=[seq(Y[i],i=n+1..2*n)];
    G:=[seq(Y[i],i=2*n+1..3*n)];
    x:= produitp(sommep(produitp(Q,F,p),3*produitp(produitp(b,R,p),G,p),p),sommep
  (produitp(R,F,p),produitp(G,Q,p),p),p) mod p;
    y:=produitp(sommep(3*produitp(P,E,p),produitp(2*produitp(a,R,p),G,p),p),
  sommep(produitp(Q,E,p),produitp(P,F,p),p),p) mod p;
    z:=produitp(a,sommep(produitp(produitp(P,Q,p),produitp(G,G,p),p),produitp
  (produitp(R,R,p),produitp(F,E,p),p),p),p) mod p;
    sommep(sommep(x,y,p),z,p);
end:
```

3.7 Law of the group $G_n^{a,b}$

In this section, we will implement the law somme defined in $G_n^{a,b}$.

Table 12. The procedure somme

1. The procedure som1

- som1:=proc(X,Y,a,b,p)

```
    local P,Q,R,n;
    P:=p1(X,Y,a,b,p);
    Q:=p2(X,Y,a,b,p);
    R:=p3(X,Y,a,b,p);
    n:=nops(P);
    if (R[1]=0) then
  P:=produitp(P,inversep(Q,p),p); Q:=produitp(Q,inversep(Q,p),p);
   R:=produitp(R,inversep(Q,p),p); else P:=produitp(P,inversep(R,p),p);
  Q:=produitp(Q,inversep(R,p),p);R:=produitp(R,inversep(R,p),p);
    end if;
  return([seq(P[i],i=1..n),seq(Q[i],i=1..n),seq(R[i],i=1..n)])
end:
```

2. The procedure som2

- som2:=proc(X,Y,a,b,p)

```
    local P,Q,R,n;
    P:=p4(X,Y,a,b,p);
    Q:=p5(X,Y,a,b,p);
```

```
    R:=p6(X,Y,a,b,p);
    n:=nops(P);
    if (R[1]=0) then
   P:=produitp(P,inversep(Q,p),p);
   Q:=produitp(Q,inversep(Q,p),p);
   R:=produitp(R,inversep(Q,p),p);
  else
   P:=produitp(P,inversep(R,p),p);
   Q:=produitp(Q,inversep(R,p),p);
   R:=produitp(R,inversep(R,p),p);
  end if;
   return([seq(P[i],i=1..n),seq(Q[i],i=1..n),seq(R[i],i=1..n)])
 end:
```

3. The procedure somme

- somme:=proc(X,Y,a,b,p)

```
    local P;
    if (projection(X)=projection(Y)) then
   P:=som2(X,Y,a,b,p)
    else
   P:=som1(X,Y,a,b,p) end if;
end:
```

4. The procedure msomme

- msomme:=proc(P,l,a,b,p)

```
    local i,point;
    point:=P;
    for i from 1 to l-1 do
    point:=somme(P,point,a,b,p):
    end do:
    return(point):
end:
```

Remark3.4 The function [m] defined by multiplying a point by an integer m is implemented through the procedure msomme.

Theorem3.5 The set ($G_n^{a,b}$, somme, Neutre, appartient, Egal) is a generic group.

Proof

1. The elements of $G_n^{a,b}$ are represented uniquely on $O((logn)^{\alpha})$ bit.
2. The operations somme, Neutre, appartient, and Egal in this group are calculated by $O((logn)^{\alpha})$ bit.
3. The Cardinal of $G_n^{a,b}$ is known.

Remark 3.6 Due to the difficulty of the discrete logarithm problem on the group $G_n^{a,b}$, it is difficult to reverse the function [m].
Solving this problem requires a non polynomial time.

3.8 Examples

1. Example 3.7:
   Let $q = 5, a = 3 + \varepsilon, b = 2 + 2\varepsilon \; and \; G = G_2^{a,b}$. Then

G = {[0, 0, 1, 0, 0, 0], [0, 1, 1, 0, 0, 0], [0, 2, 1, 0, 0, 0],[0, 3, 1, 0, 0, 0], [0, 4, 1, 0, 0, 0], [1, 0, 1, 4, 1, 0], [1, 0, 4, 1, 1, 0], [1, 1, 1, 2, 1, 0], [1, 1, 4, 3, 1, 0],[1, 2, 1, 0, 1, 0], [1, 2, 4, 0, 1, 0], [1, 3, 1, 3, 1, 0],[1, 3, 4, 2, 1, 0], [1, 4, 1, 1, 1, 0], [1, 4, 4, 4, 1, 0], [2, 0, 1, 2, 1, 0], [2, 0, 4, 3, 1, 0], [2, 1, 1, 2, 1, 0], [2, 1, 4, 3, 1, 0], [2, 2, 1, 2, 1, 0], [2, 2, 4, 3, 1, 0],[2, 3, 1, 2, 1, 0], [2, 3, 4, 3, 1, 0], [2, 4, 1, 2, 1, 0],[2, 4, 4, 3, 1, 0]}.
   The cardinal of G is 25, G is cyclic generated by [1, 4, 1, 1, 1, 0].

2. Example 3.8:

   Let $q = 991, a = 291 + 213\varepsilon + 514\varepsilon^2, b = 637 + 341\varepsilon + 729\varepsilon^2 \; and \; G = G_3^{a,b}$.
   Then, the cardinal of G is 1020730.
   G is cyclic generated by P=[919,110,981,726,433,470,1,0,0].
   Let Q=[395,706,113,692,506,128,1,0,0], then $log_P Q = 108322$.

References

1. A. Chillali "Ellipic cuvre over ring $\mathbb{F}_q[\varepsilon]$; $\varepsilon^n = 0$ " International Mathematical Forum, Vol. 6, no . 31, 1501-1505, 2011.
2. A. Chillali, "Identification methods over $E_n^{a,b}$", In Proceedings of the 2011 international conference on Applied computational mathematics (ICACM'11), Vladimir Vasek, Yuriy Shmaliy, Denis Trcek, Nobuhiko P. Kobayashi, and Ryszard S. Choras (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 133-138.2011.
3. A. Chillali, "Cryptography Over Elliptic Curve Of The Ring $\mathbb{F}_q[\varepsilon]$; $\varepsilon^4 = 0$", World Academy of Science, Engineering and Technology 78 2011, pages 847-850, 2011.
4. Koblitz N, " Elliptic Curve Cryptosystems, Mathematics of Computation," 48 203-209, 1987.
5. M. Virat, "Courbe elliptique sur un anneau et applications cryptographiques," Thèse Docteur en Sciences, Nice-Sophia Antipolis 2009.