KRY: Wahlfach Kryptologie
Serie 13: Kryptographie auf elliptischen Kurven

Aufgabe 1 (T)
Wir betrachten die elliptische Kurve $E: y^2 = x^3 + 2x + 5$ über GF(13).
Der Punkt $P(2,2)$ hat die Ordnung 12 und erzeugt die elliptische Kurve.
Alice möchte Bob mit Hilfe des El Gamal Systems die Nachricht $M(5,7)$ übermitteln. Sie wählt als geheimen Schlüssel $k_A = 5$. Bob wählt $k_B = 4$.

(a) Bestimmen Sie das Chiffrat von M mithilfe von *elladd* und *ellpow*.

(b) Entschlüsseln Sie das Chiffrat mithilfe von *elladd* und *ellpow*.

Aufgabe 2 (T)
Wir betrachten die Kurve E: $y^2 = x^3 + 4x + 8$ über GF(23677).

(a) Stellen Sie die Nachricht $m = 1101$ als Punkt M der elliptischen Kurve dar, indem Sie einen Bitshift um 4 Positionen vornehmen.

(b) Geben Sie an, wie aus dem Punkt M die Nachricht m rekonstruiert wird.

Aufgabe 3 (T)
Alice will mit ECDSA ein Dokument bzw. dessen Hash-Wert $h(m) = 121$ signieren. Sie verwendet dazu die elliptische Kurve $E: y^2 = x^3 + 8x + 102$ über $GF(179)$ und daraus den Punkt $P = (73, 60)$ mit Ordnung $|P| = 167$. Ihr geheimer Schlüssel lautet $d = 37$ und zum Signieren wählt sie die Zufallszahl $k = 94$.

(a) Bestimmen Sie den öffentlichen Schlüssel Q von Alice.

(b) Bestimmen Sie die Signatur von $h(m)$.

(c) Überprüfen Sie die in (b) berechnete Signatur. Sie dürfen dabei voraussetzen, dass der Hash-Wert des Dokumentes bei der Überprüfung auch wieder $h(m) = 121$ ergibt.

(d) Alice signiert nun mit den gleichen Parametern, insbesondere mit dem gleichen k, ein zweites Dokument mit Hash-Wert $h(\tilde{m}) = 83$. Zeigen Sie, wie ein Angreifer, der die beiden Hash-Werte und die beiden Signaturen mitbekommen hat, daraus den geheimen Schlüssel von Alice berechnen kann.

1.
```
E = ellinit([0,2,0,5,0]*Mod(1,13));
P = [2,2]*Mod(1,13);
M = [5,7]*Mod(1,13);
k_A = 5;
k_B = 4;
```

a)
```
k_A_P = ellmul(E, P, k_A);
k_B_P = ellmul(E, P, k_B);

C = [k_B_P, elladd(E, M, k_A_P)];
gp > C ==> [[Mod(2, 13), Mod(11, 13)], [Mod(5, 13), Mod(7, 13)]]
```

b)
```
k_A_k_B_P = ellmul(E, k_B_P, k_A);

M_prime = elladd(E, C[2], -k_A_k_B_P);
gp > M_prime ==> [Mod(5, 13), Mod(7, 13)]
```

2.
```
p = 23677;
E = ellinit([0,4,0,8,0], p);
m = 1101;
```

a)
```
M_x = Mod(m, p)^4;
M_y_squared = lift(Mod(M_x^3 + 4*M_x + 8, p));

M = [M_x, M_y_squared];
gp > M ==>  [Mod(16082, 23677), 10493]
```

b)
```
x_M = M[1];
y_M_squared = M[2];
m_re = Mod(y_M_squared, p)^( (p+1)/4 );

gp > m_re ==> Mod(22404, 23677)
```

3.

```
p = 179;
E = ellinit([0, 8, 0, 102, 0], p);
```

a)

```
Q = lift(37 * P);
gp > Q ==> [2701, 2220]
```

b)

```
h_m = 121;
k = 94;

G = Mod(0, p);
R = lift(k * P);

r = Mod(R[1], 167);

s = Mod((h_m + 37 * r) / k, 167);
gp > s ==> Mod(128, 167)
```

c)

```
w = lift(1 / s);
u1 = Mod(h_m * w, 167);
u2 = Mod(r * w, 167);

V = u1 * P + u2 * Q;

valid_signature = Mod(V[1], 167) == r;
gp > valid_signature ==> 1 => true
```

d)

```
h_m1 = 121;
h_m2 = 83;
s1=128
s2=21

delta_s = Mod(s2 - s1, 167);
delta_h = Mod(h_m2 - h_m1, 167);

inv_delta_s = lift(1 / delta_s);
gp > inv_delta_s ==> 103

k_inv = Mod(delta_h * inv_delta_s, 167);
gp > k_inv ==> Mod(94, 167)

d_attacker = Mod((r * s1 - h_m1) * k_inv, 167);
gp > d_attacker ==> Mod(102, 167)
```