Модуль расчета ЭЦП

```
ECDSAGenerate[p0_, a0_, b0_, c0_, P0_, q0_, file0_, secretKey0_] := Module[
  {p = p0, a = a0, b = b0, ci = c0, P = P0, q = q0, file = file0, secretKey = secretKey0, randomKey, R, r, hash, randomKeyInv, s},
  randomKey = RandomInteger[{1, q - 1}];
  R = MulByK[p, a, b, ci, P, randomKey];
  r = Mod[R[[1]], q];
  hash = FileHash[file, "MD5"];
  randomKeyInv = PowerMod[randomKey, -1, q];
  s = Mod[randomKeyInv * (hash + secretKey * r), q];
  {r, s}
 ]
```

Модуль верификации ЭЦП

```
ECDSAVerificate[p0_, a0_, b0_, c0_, P0_, q0_, file0_, Q0_, sign0_] := Module[
  {p = p0, a = a0, b = b0, ci = c0, P = P0, q = q0, file = file0, Q = Q0, sign = sign0, W, hash, U1, U2, U1P, U2Q, R, r, s},
  r = sign[[1]]; s = sign[[2]];
  If[(1 < r < q - 1) && (1 < s < q - 1),
   W = PowerMod[s, -1, q];
   hash = FileHash[file, "MD5"];
   U1 = Mod[W * hash, q];
   U2 = Mod[W * r, q];
   U1P = MulByK[p, a, b, ci, P, U1];
   U2Q = MulByK[p, a, b, ci, Q, U2];
   R = EllipticAdd[p, a, b, ci, U1P, U2Q];
   If[R[[1]] = r,
    Print["Подпись принята"]
    , Print["Подпись отклонена"]
   ],
   Print["Условие интервала не выполнено"]
  ];
 ]
```