On the feasibility of an ECDLP algorithm

Sergey Grebnev*

HSE Tikhonov Moscow Institute of Electronics and Mathematics
(MIEM HSE)

Saturday 28th April, 2018

Abstract

We study the properties of an algorithm for solving the elliptic curve discrete logarithm problem presented by A. Yu. Nesterenko at the CTCrypt 2015 session. We show that for practically important instances of the problem its average complexity is not less than that of Pollard's ρ-method.

Keywords: elliptic curves, discrete logarithm problem, multiplicative orders, Pollard's lambda, Pollard's rho.

1 Introduction

For an elliptic curve E given over the field $GF(p)$ by the equation

$$y^2 = x^3 + ax + b, \tag{1}$$

we define the *discrete logarithm problem*: for given $P, Q \in E(GF(p))$, find

$$x : 0 \leq x \leq \#E(GF(p)) \text{ such that } Q = xP, \tag{2}$$

if one exists.

The problem is believed to be computationally hard: the best generic algorithm, Pollard's ρ-method proposed in [4], as well as its efficient paralleling [7], both have the asymptotic complexity of $O(\#E(GF(p)))$ elliptic

*sgrebnev@hse.ru

curve additions. Therefore, operations in the group of points of an elliptic curve allow to implement a number of cryptographic primitives, e.g. digital signature schemes such as GOST R 34.10-2012 a.k.a. ECRDSA, ECDSA, EdDSA etc.

In 2016 A. Yu. Nesterenko proposed an algorithm which exploits the properties of the multiplicative group modulo q, where q is the prime order of a subgroup of the group of points of an elliptic curve, to speed up the computation of the discrete logarithm problem for x with a small multiplicative order modulo $\#E(GF(p))$.

We proceed with the description of the algorithm, study its complexity and briefly discuss its applicability to the real-world cryptosystems.

2 Discrete logarithm using the multiplicative properties of the variable

The algorithm proposed in [3] exploits the properties of the multiplicative group modulo a prime q. We recall the algorithm following the original paper.

Without loss of generality we suppose that $\#E(GF(p)) = q$, with q prime.

For an integer x coprime to a prime q its *order* modulo q is defined as an integer r such that

$$x^r \equiv 1 \pmod{q} \quad \text{and} \quad x^l \not\equiv 1 \pmod{q}, \quad \text{for each} \quad 1 \leq l < r.$$

We have that $r|(q-1)$. Hence, there exist α and n such that

$$\alpha^n \equiv x \pmod{q}, \quad 0 \leq n < r. \tag{3}$$

We have $\alpha \equiv g^{\frac{q-1}{r}} \pmod{q}$, where g is a primitive root modulo q.

Now we use equation (3) to rewrite (2) as

$$\alpha^n P = Q, \tag{4}$$

so by finding n we can determine x.

We put $h = \lceil \sqrt{r} \rceil$ and write n as $n_1 h - n_0$, where $0 \le n_0 < h$, $0 < n_1 \le h$. Then we have for (4)

$$(\alpha^h)^{n_1} P = \alpha^{n_0} Q;$$

now, by Hellman's method [2], we determine n_0, n_1 to find n and x. The procedure requires at most $2h$ scalar multiplications and $O(\sqrt{q})$ memory.

For practical evaluation the paper [3] proposes a modification of Pollard's λ-method.

We fix an integer $s = \lceil \log_2 r \rceil$, choose random $\xi_0, \ldots, \xi_{s-1}$ such that $0 < \xi_i < r$, $i = 0, \ldots, s-1$, and define a map $f : E \to E$ in the following manner. For a point $R = (x_R, y_R) \in E$ let

$$f(R) = \zeta_i R, \quad \text{where} \quad i \equiv x_R \pmod{s}, \quad \zeta_i \equiv a^{\xi_i} \pmod{q}. \tag{5}$$

We may consider f as a random map.

At the stage I of the algorithm we fin an integer β and construct the orbit of P under the map f, that is, the finite sequence

$$R_{k+1} = f(R_k), \quad R_0 = P, \quad k = 0, 1 \ldots, \beta. \tag{6}$$

We have that

$$R_{k+1} = \zeta_k R_k = \zeta_k \zeta_{k-1} R_{k-1} = \cdots = \mu_{k+1} P,$$

where $\mu_{k+1} \equiv \prod_{j=0}^{k} \zeta_j \pmod{q}$, and ζ_j are pseudorandom values from the set defined by (5).

Starting from an index k_0, we store the set S of points $R_{k_0+1}, \ldots, R_{k_0+\beta}$ and the corresponding values $\mu_{k_0+1}, \ldots, \mu_{k_0+\beta}$. We call these points ¡¡traps¿¿, following [4], and S is the ¡¡trap-set¿¿.

At the stage II of the algorithm we construct the orbit of a point $U_0 = \alpha^{\xi} Q$ for a random ξ such that $0 < \xi < r$, under the map f:

$$U_{k+1} = f(U_k), \quad U_0 = \alpha^{\xi} Q, \quad k = 0, 1, \ldots, \quad 0 < \xi < r. \tag{7}$$

Now we also have that the following relationships hold:

$$U_{k+1} = \zeta_k U_k = \zeta_k \zeta_{k-1} U_{k-1} = \cdots = \nu_{k+1} Q,$$

where $\nu_{k+1} \equiv \alpha^{\xi} \prod_{j=0}^{k} \zeta_j \pmod{q}$.

Now for every point U_k we check whether $U_k \in S$. If for an index j we have that

$$\mu_j P = P_j = U_k = \nu_k Q,$$

we conclude that $\mu_j \equiv \nu_k x \pmod{q}$, and hence

$$x \equiv \mu_j \nu_k^{-1} \pmod{q}.$$

If we have failed to find a trap among all the indexes $k \leq h$, we restart the stage II of the algorithm with another random value ξ such that $0 < \xi < r$.

3 Complexity analysis

Recall that for Pollard's ρ-method we have (see [4, 6]) the following complexity estimate independent of x:

$$\sqrt{\frac{\pi q}{2\#\mathfrak{G}}}, \tag{8}$$

elliptic curve points additions, where $\mathfrak{G}$ is the group of efficiently computable automorphisms.

The automorphisms may be exploited for the method described above just like the Pollard's ρ: we define the map f on the equivalency classes $E(GF(p))/\mathfrak{G}$ and fix a specific representative of an equivalence class to store traps.

Thus, for a given unknown x such that $\operatorname{ord}_q x = r$ the algorithm described in section 2 has the complexity

$$T(x) = \sqrt{\frac{\pi \operatorname{ord}_q x}{2\#\mathfrak{G}}} \cdot \log_2 q \tag{9}$$

where the factor $\log_2 q$ is determined by the complexity of the scalar multiplication $\alpha^{\xi_i} P$, see [3].

We suppose that the group $\mathfrak{G}$ is generated by the map $\psi : P \mapsto -P$ and thus $\#\mathfrak{G} = 2$. This is always the case for a non-anomalous, non-supersingular Weierstrass curve (1) defined over $GF(p)$ with $p > 3$, where

$1728\frac{4a^3}{4a^3+27b^2} \neq 0, 1728 \pmod{p}$. In particular, this assumption holds for the family of elliptic curves allowed by the national standard [1].

Now we estimate the average complexity S of the algorithm:

$$S = \frac{1}{q-1} \sum_{0<x<q} T(x), \tag{10}$$

where $T(x)$ is the complexity of determining the specific discrete logarithm x.

Note that we may omit the term corresponding to $q-1$, since we obviously have $(q-1)P = -P$, and thus $T(q-1) = 0$ elliptic curve additions.

Now write (10) as

$$\begin{aligned} S &= \frac{1}{q-1} \sum_{r|\varphi(q)} (\#\{x : \operatorname{ord}_q x = r\}) T(r) = \\ &= \frac{\sqrt{\pi}\log_2 q}{2(q-1)} \left(\sum_{r|\varphi(q)} \varphi(r)\sqrt{r} \right) \end{aligned} \tag{11}$$

We note that the exact value S in (11) depends on the factorization of $\varphi(q)$:

$$\varphi(q) = q - 1 = r_1^{\alpha_1} \cdot \dots \cdot r_k^{\alpha_k}.$$

We may, however, obtain a lower bound for (11). Recall that we have q prime, which is the case for most cryptographic applications. Consider the sum

$$\frac{\sqrt{\pi}\log_2 q}{2(q-1)} \left(\sum_{r|\varphi(q)} \varphi(r)\sqrt{r} \right). \tag{12}$$

For prime q we have $\varphi(q) = q - 1$ is divided by $(q-1)/2$. Then the largest term of (11) relating to $(q-1)/2$ equals to

$$\frac{\sqrt{\pi}\log_2 q}{2(q-1)} \cdot \varphi((q-1)/2)\sqrt{(q-1)/2}.$$

Since for any integer $l \geq 3$ we have (see [5]) the inequality

$$l > \varphi(l) > \frac{\ln 2}{2} \cdot \frac{l}{\ln l},$$

we estimate (12) in the following manner:

$$\frac{\sqrt{\pi}\log_2 q}{2(q-1)}\cdot\frac{\ln 2}{2}\cdot\frac{(q-1)/2}{\ln(q-1/2)}\cdot\sqrt{(q-1)/2}\geq\frac{\sqrt{\pi}}{4\sqrt{2}}\cdot\frac{\log_2 q}{\log_2(q-1)}\cdot\sqrt{q-1}=O(\sqrt{q});$$

thus, we deduce that the lower asymptotical bound for the average complexity of the algorithm is the same as of Pollard's original method.

4 Conclusion

We conclude that for cryptographically significant cases the average complexity of the method for solving the elliptic curve discrete logarithm problem proposed in [3] is at least as large as that of Pollard's original method. This fact makes it unfeasible for attacking most of the real-word cryptosystems, including ECRDSA [1], where $2^{508} < q < 2^{512}$.

References

[1] *GOST R 34.10-2012. Information technology. Cryptographic protection. The processes of generation and validation of digital signatures (in Russian)*, Moscow:Standartinform, 2013.

[2] Hellman M., "A cryptanalytic time-memory trade-off", *IEEE Trans. Inform. Theory*, **IT-26**:4 (1980), 401-406.

[3] A. Yu. Nesterenko, "Some remarks on the elliptic curve discrete logarithm problem", *Mat. Vopr. Kript*, **7**:2 (2016), 115–120.

[4] Pollard J.M., "Monte Carlo methods for index computation (mod p)", *Math. Comp.*, **32**:143 (1978), 918–924.

[5] Sandor J., Mitrinovic D., Crstici B., *Handbook of number theory I*, Springer, 2005.

[6] Teske E., "Square-root algorithms for the discrete logarithm problem (a survey)", *Public-key cryptography and computational number theory (Warsaw, 2000)*, Springer, 2000, 283-301.

[7] van Oorschot P.C., Wiener M.J., "Parallel collision search with cryptanalytic applications", *J. Cryptology*, **12**:1 (1999), 1–128.