COMPRESSION FOR TRACE ZERO SUBGROUPS OF ELLIPTIC CURVES

A. SILVERBERG

Abstract. We give details of a compression/decompression algorithm for points in trace zero subgroups of elliptic curves over $\mathbb{F}_{q^r}$, for $r = 3$ and 5.

1. Introduction

"Classical" point compression on elliptic curves consists of taking a point (x_0, y_0) on an elliptic curve $y^2 = F(x)$ and dropping its y-coordinate. To decompress (up to a sign ambiguity in the y-coordinate), use x_0 and the equation of the elliptic curve to solve for y_0.

In [2] (see also §4.2 of [3]), Rubin and Silverberg gave a method for compressing and decompressing points on trace zero subgroups of elliptic curves over extension fields of degree 3 or 5. In the current paper we give a survey of that method and fill in some of the details in the algorithm, that space limitations did not allow in [2].

In [2, 3] we concentrated on the case of supersingular elliptic curves, having in mind applications to pairing-based cryptography, including short signatures. In this paper we abstract out the compression and decompression technique, and deal with a general elliptic curve over a finite field $\mathbb{F}_{q^r}$.

In §2 we give an algorithm for compressing and decompressing points in the trace zero subgroup of an elliptic curve over $\mathbb{F}_{q^r}$, which we can make practical when $r = 3$ or 5. We compress by a factor of $r/(r-1)$. We demonstrate this in the case where $r = 3$ in §3, and in the case where $r = 5$, q is a power of 3, and E is one of the elliptic curves $y^2 = x^3 - x \pm 1$, in §§4–5. Note that these $r = 5$ examples are useful for pairing-based cryptography, since these curves are supersingular curves of security parameter 6 (the maximal security parameter for a supersingular elliptic curve), whose primitive (i.e., trace zero) subgroups over $\mathbb{F}_{q^5}$ have security parameter 7.5.

In §7 we explain the algorithm.

The results reported on in this paper were obtained jointly with Karl Rubin, whom the author thanks for help with the paper. The author also thanks NSF and NSA for partial support.

2. Compression and decompression of points in trace zero subgroups

In this section we present the idea of the algorithm. We make it practical in §§3–5, and explain why it works in §7.

Let

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

Key words and phrases. elliptic curves, compression.

be an elliptic curve over a finite field $\mathbb{F}_q$, and let O_E denote its identity element. Suppose r is odd and let σ be a generator of $\mathrm{Gal}(\mathbb{F}_{q^r}/\mathbb{F}_q)$. For $Q \in E(\mathbb{F}_{q^r})$, let

$$\mathrm{Tr}(Q) = \mathrm{Tr}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(Q) = \sum_{i=0}^{r-1} \sigma^i(Q).$$

Let

$$A_0 = \{Q \in E(\mathbb{F}_{q^r}) : \mathrm{Tr}(Q) = O_E\},$$

the "trace zero subgroup" of $E(\mathbb{F}_{q^r})$. Write $\mathbb{F}_{q^r} = \mathbb{F}_q[z]/f(z)\mathbb{F}_q[z]$ with $f(z) \in \mathbb{F}_q[z]$ irreducible and of degree r.

Compression: The input is a point $P = (s,t) \in A_0$.

(i) Write $s = \sum_{i=0}^{r-1} s_i z^i$, i.e., write s with respect to the basis $\{1, z, \dots, z^{r-1}\}$ for $\mathbb{F}_{q^r}$ over $\mathbb{F}_q$.
(ii) Output $(s_1, \dots, s_{r-1})$ (i.e., drop t and the first coordinate s_0).

Decompression: The input is $(s_1, \dots, s_{r-1}) \in \mathbb{F}_q^{r-1}$. The output will be the point P (i.e., we will reconstruct $s_0 \in \mathbb{F}_q$ and $t \in \mathbb{F}_{q^r}$), up to some small ambiguity.

(i) Let $c = S + \sum_{i=1}^{r-1} s_i z^i$. Compute the (monic) characteristic polynomial $g(X)$ of the linear transformation on $\mathbb{F}_{q^r}$ given by multiplication by c, where the variable S is treated as an element of $\mathbb{F}_q$ (in other words, we have $g(X) = \prod_{i=0}^{r-1}(X - \sigma^i(c))$; the equation $f(z) = 0$ can be used to compute the matrix for multiplication by c with respect to the basis $\{1, z, \dots, z^{r-1}\}$ for $\mathbb{F}_{q^r}$ over $\mathbb{F}_q$).
(ii) Equate coefficients of like powers of X in equation (2) below to obtain r equations in the r unknowns $S, \alpha_0, \dots, \alpha_{(r-1)/2}, \beta_0, \dots, \beta_{(r-5)/2}$, and solve that system of equations.

$$\begin{aligned} g(X) = \big(X^{(r-3)/2} + \sum_{i=0}^{(r-5)/2} \beta_i X^i\big)^2 (X^3 + a_2X^2 + a_4X + a_6) \\ + \big(X^{(r-3)/2} + \sum_{i=0}^{(r-5)/2} \beta_i X^i\big)\big(\sum_{i=0}^{(r-1)/2} \alpha_i X^i\big)(a_1X + a_3) - \big(\sum_{i=0}^{(r-1)/2} \alpha_i X^i\big)^2 \end{aligned} \quad (2)$$

The solutions for S give s_0, and thus s, up to some ambiguity.
(iii) Use (1) to solve for t, up to some ambiguity (discarding any candidate s_0's that do not produce a $t \in \mathbb{F}_q$).

Note that any ambiguity can be resolved by transmitting extra bits that allow the decompressor to determine which solution of the system of equations to choose (and which of two possibilities to choose for t).

The compression algorithm is clearly efficient, since it just consists of dropping t and one coordinate of s. We found a way to make the decompression algorithm practical when $r = 3$ or 5. We demonstrate this below.

3. An example: $E(\mathbb{F}_{q^3})$

Let $r = 3$, and assume that the characteristic is not 3. Then we can take the irreducible polynomial defining the degree three extension $\mathbb{F}_{q^3}$ to be of the form $f(z) = z^3 + r_1z + r_0$ with $r_i \in \mathbb{F}_q$, and we can take $a_2 = 0$.

Then the characteristic polynomial of multiplication by $S + s_1 z + s_2 z^2$ is

$$g(X) = X^3 + (2s_2r_1 - 3S)X^2 + (3s_1s_2r_0 + s_1^2r_1 + s_2^2r_1^2 - 4s_2r_1S + 3S^2)X \\ + s_1^3r_0 - s_2^3r_0^2 + s_1s_2^2r_0r_1 - 3s_1s_2r_0S - s_1^2r_1S - s_2^2r_1^2S + 2s_2r_1S^2 - S^3.$$

The difference of the two sides of (2) is now

$$(3S - \alpha_1^2 + a_1\alpha_1 - 2s_2r_1)X^2 \\ + (\alpha_0(a_1 - 2\alpha_1) + a_3\alpha_1 + 3S^2 - 3s_1s_2r_0 - s_1^2r_1 + 4Ss_2r_1 - s_2^2r_1^2 + a_4)X \\ + a_6 + S^3 + a_3\alpha_0 - \alpha_0^2 - s_1^3r_0 + 3Ss_1s_2r_0 \\ + s_2^3r_0^2 + Ss_1^2r_1 - 2S^2s_2r_1 - s_1s_2^2r_0r_1 + Ss_2^2r_1^2. \quad (3)$$

Setting the coefficient of the quadratic term of (3) equal to 0 and solving for S gives

$$S = (\alpha_1^2 - a_1\alpha_1 + 2s_2r_1)/3.$$

Substituting into (3), setting the coefficient of the linear term equal to 0, and solving for α_0 gives

$$\alpha_0 = (3(a_4 + a_3\alpha_1 - 3s_1s_2r_0 - s_1^2r_1) - a_1^2\alpha_1^2 + 2a_1\alpha_1^3 - \alpha_1^4 + s_2^2r_1^2)/(3(2\alpha_1 - a_1)).$$

Substituting these into (3) and setting the constant term equal to 0 gives a degree 8 polynomial in $\mathbb{F}_q[\alpha_1]$, which can be rewritten as $G(\alpha_1^2 - a_1\alpha_1)$ with

$$G(t) = t^4 + a_1^2t^3 + 3(3(a_1a_3 + 2a_4 + 6s_1s_2r_0 + 2s_1^2r_1) - 2s_2^2r_1^2)t^2 \\ + (27(a_3^2 + a_1^2s_1s_2r_0 + 4(a_6 - s_1^3r_0 + s_2^3r_0^2 + s_1s_2^2r_0r_1)) \\ + 3r_1(3a_1^2s_1^2 + 24s_1^2s_2r_1 - a_1^2s_2^2r_1) + 8s_2^3r_1^3)t \\ + 27(a_1^2a_6 - a_1a_3a_4 - a_4^2 - a_1^2s_1^3r_0 + a_1^2s_2^3r_0^2 + a_1a_3s_1^2r_1 + a_1^2s_1s_2^2r_0r_1 - s_1^4r_1^2 \\ + 2a_4s_1^2r_1 + 3a_1a_3s_1s_2r_0 + 6a_4s_1s_2r_0 - 9s_1^2s_2^2r_0^2 - 6s_1^3s_2r_0r_1 + 2s_1s_2^3r_0r_1^2) \\ + 18(a_1^2s_1^2s_2r_1^2 - a_4s_2^2r_1^2 + s_1^2s_2^2r_1^3) + 2a_1^2s_2^3r_1^3 - 9a_1a_3s_2^2r_1^2 - 3s_2^4r_1^4.$$

Compute the roots of $G(t)$. Let $s_0 = S = (R + 2s_2r_1)/3$, where R is a root of $G(t)$.

Transmit three extra bits to determine P with no ambiguity (two to determine which root of G to choose, and one to determine t).

3.1. **Characteristic 2.** When $q = 2^n$ with n not divisible by 3, we may take $r_0 = r_1 = 1$. If further $a_1 = 0$, then decompression becomes easier if the compression algorithm drops s_1 or s_2, rather than s_0. In that case, setting the coefficient of the quadratic term of (3) equal to 0 yields the equation $s_0 = \alpha_1^2$. Solving for α_1 amounts to taking a square root, which is just an exponentiation in characteristic 2. Using the linear term of (3) one obtains

$$s_1^2 + s_1s_2 + s_2^2 + s_0^2 + a_3\alpha_1 + a_4 = 0,$$

a quadratic polynomial in s_1 (or s_2), and solving a quadratic equation is also easy in characteristic 2. This is especially useful in the (supersingular) cases $y^2 + y = x^3 + x$ and $y^2 + y = x^3 + x + 1$ considered in §5.2 of [2].

3.2. **Characteristic ≥ 5.** When the characteristic is at least 5, we may take a model for E with $a_1 = a_3 = 0$. Then either $\alpha_1 = 0$ and $S = 2s_2r_1/3$, or α_1 is a solution of

$$\begin{aligned}\alpha_1^8 + \alpha_1^4(18a_4 + 54s_1s_2r_0 + 18s_1^2r_1 - 6s_2^2r_1^2) \\ + \alpha_1^2(108(a_6 - s_1^3r_0 + s_2^3r_0^2 + s_1s_2^2r_0r_1) + 8(9s_1^2s_2r_1^2 + s_2^3r_1^3)) \\ + 27(2s_1s_2^3r_0r_1^2 - a_4^2 + 6a_4s_1s_2r_0 - 9s_1^2s_2^2r_0^2 + 2a_4s_1^2r_1 - 6s_1^3s_2r_0r_1 - s_1^4r_1^2) \\ + 18s_2^2(s_1^2r_1^3 - a_4r_1^2) - 3s_2^4r_1^4 = 0.\end{aligned}$$

4. An example: $E(\mathbb{F}_{q^5})$ in characteristic 3

Consider the elliptic curve $E_1 : y^2 = x^3 - x - 1$ over $\mathbb{F}_q$ where $q = 3^n$ with n not divisible by 5. A similar computation can be done for $E_2 : y^2 = x^3 - x + 1$.

The security parameter for the elliptic curve is 6. However, using our compression algorithm on the trace zero subgroup of $E_1(\mathbb{F}_{q^5})$ boosts the security parameter up to 7.5, as shown in [2], thereby obtaining greater efficiency for security comparable to that of the full group $E_1(\mathbb{F}_{q^5})$.

Write $\mathbb{F}_{q^5} = \mathbb{F}_q[z]/f(z)\mathbb{F}_q[z]$ with $f(z) = z^5 - z + 1$. Define $b_0, \cdots, b_4 \in \mathbb{F}_q[S]$ by

$$g(X) = X^5 + \sum_{i=0}^{4} b_i X^i$$

where

$$g(X) = X^5 + (S - s_4)X^4 + (s_2^2 - s_1s_3 - s_2s_3 - s_1s_4 + s_4S + S^2)X^3 + \cdots$$

is the characteristic polynomial of multiplication by $S + \sum_{i=1}^{4} s_i z^i$ on $\mathbb{F}_{q^5}$. Now write

$$X^5 + \sum_{i=0}^{4} b_i X^i = (X + \beta_0)^2(X^3 - X - 1) - (\alpha_2 X^2 + \alpha_1 X + \alpha_0)^2.$$

Taking the difference of the two sides gives

$$\begin{aligned}(\alpha_2^2 + \beta_0 + b_4)X^4 + (1 - \alpha_1\alpha_2 - \beta_0^2 + b_3)X^3 \\ + (\alpha_1^2 - \alpha_0\alpha_2 - \beta_0 + b_2 + 1)X^2 + (\beta_0^2 - \beta_0 - \alpha_0\alpha_1 + b_1)X + \alpha_0^2 + \beta_0^2 + b_0.\end{aligned}$$

Setting the coefficient of the degree four term equal to 0 and solving for β_0 gives

$$\beta_0 = -\alpha_2^2 - b_4.$$

Setting the coefficient of the cubic term equal to 0 and solving for α_1, one obtains

$$\alpha_1 = (1 - \alpha_2^4 + \alpha_2^2 b_4 - b_4^2 + b_3)/\alpha_2.$$

Setting the coefficient of the quadratic term equal to 0 and solving for α_0 gives

$$\alpha_0 = \frac{\alpha_2^8 + \alpha_2^6 b_4 + b_4^4 + b_4^2(1 + b_3) + (1 + b_3)^2 + \alpha_2^4(b_3 - 1) + \alpha_2^2(1 + b_4^3 - b_4 b_3 + b_2)}{\alpha_2^3}.$$

Setting the constant (respectively, coefficient of the linear) term equal to 0 gives polynomials in α_2^2, so replace α_2^2 with a new variable, w. The resulting equations

are, respectively, $p_1(w) = 0$ and $p_2(w) = 0$ where

$$\begin{aligned} p_1(w) = w^8 - b_4 w^7 + (1 + b_4^2 - b_3) w^6 + (b_4 - b_4^3 - b_2) w^5 + (b_4 - b_4^2 + b_4^4 - b_3 - b_4 b_2) w^4 \\ + (1 - b_4 + b_4^2 - b_4^5 - b_3 + b_4^3 b_3 + b_2 - b_3 b_2 + b_0) w^3 \\ + (-1 + b_4^2 - b_4^3 + b_4^4 + b_4^6 + b_3 + b_4 b_3 - b_3^2 - b_3^3 - b_2 - b_4^3 b_2 + b_4 b_3 b_2 + b_2^2) w^2 \\ + (-1 - b_4^2 - b_4^3 - b_4^4 - b_4^5 - b_4^7 + b_3 + b_4 b_3 - b_4^2 b_3 - b_4^3 b_3 - b_3^2 \\ - b_4 b_3^2 + b_4 b_3^3 - b_2 - b_4^2 b_2 - b_4^4 b_2 + b_3 b_2 - b_4^2 b_3 b_2 - b_3^2 b_2) w \\ + 1 - b_4^2 - b_4^6 + b_4^8 + b_3 - b_4^6 b_3 + b_3^3 - b_4^2 b_3^3 + b_3^4, \end{aligned}$$

$$\begin{aligned} p_2(w) = w^6 - w^4 + (-1 - b_4 - b_4^3 + b_2) w^3 + (-1 + b_4^2 - b_3 - b_4 b_2 + b_1) w^2 \\ + (-1 - b_4 + b_4^2 + b_4^3 - b_3 - b_4 b_3 - b_2 + b_4^2 b_2 - b_3 b_2) w - 1 + b_4^6 - b_3^3. \end{aligned}$$

Taking the resultant of p_1 and p_2 eliminates the variable w, and gives a (degree 27) polynomial $h(S) \in \mathbb{F}_q[S]$ that has s_0 as a root. We observe that the polynomial $h(S)$ is of the form $H(S^3 - S)$ for a certain degree 9 polynomial $H(S) \in \mathbb{F}_q[S]$ (this follows from the fact that $(x, y) \mapsto (x+1, y)$ is an automorphism of the curve E_1, of order 3), and this simplifies finding the roots of h. Transmitting 6 extra bits allows one to recover s_0 and t exactly, with no ambiguity; 5 bits determine which root to choose (of at most 27), and one bit determines the sign of the y-coordinate t.

Remark 1. Using the number theory software package KASH to compute the resultant and find its roots, on a desktop computer decompression takes about 300 ms using E_1 with $q = 3^{19}$ and takes about 1.5 seconds using E_2 with $q = 3^{43}$ (these are good parameters, in the sense that they are in a cryptographically useful range and the order of the trace zero subgroup is divisible by a large prime). This could be sped up by writing a dedicated program.

Remark 2. One could express p_1 and p_2 as polynomials in w and the b_i's (or s_i's), and compute the resultant with the b_i's (or s_i's) viewed as variables. The computation of the resultant would need to be done only once, but the resultant polynomial computed this way is so large that evaluating it each time on particular b_i's or s_i's takes longer than computing the resultant anew each time with particular values for the b_i's or s_i's.

5. An explicit example: $E_1(\mathbb{F}_{3^{95}})$

We give details of the above algorithm for compressing points in the trace zero subgroup of $E_1(\mathbb{F}_{q^5})$, with $E_1 : y^2 = x^3 - x - 1$ and $q = 3^{19}$. We view $\mathbb{F}_q$ as $\mathbb{F}_3(\eta)$ where $\eta^{19} - \eta^2 + 1 = 0$, we view $\mathbb{F}_{3^5}$ as $\mathbb{F}_3(z)$ where $z^5 - z + 1 = 0$, and we view $\mathbb{F}_{q^5}$ as $\mathbb{F}_3(z + \eta)$ $(= \mathbb{F}_3(z, \eta))$.

Suppose that the compressor sends $(s_1, s_2, s_3, s_4) \in \mathbb{F}_q^4$ to the decompressor, where

$$\begin{aligned} s_1 &= \eta^{18} + \eta^{17} - \eta^{16} - \eta^{13} - \eta^{10} + \eta^9 + \eta^7 + \eta^6 + \eta^5 + \eta^4 + \eta^2 + \eta - 1, \\ s_2 &= \eta^{17} + \eta^{16} - \eta^{13} - \eta^{12} - \eta^{11} - \eta^8 + \eta^7 - \eta^5 + \eta^4 + \eta^2, \\ s_3 &= -\eta^{17} + \eta^{16} - \eta^{15} + \eta^{14} + \eta^{13} + \eta^{12} - \eta^{10} + \eta^7 - \eta^4 + \eta - 1, \\ s_4 &= -\eta^{18} - \eta^{16} - \eta^{14} - \eta^{13} + \eta^{12} + \eta^{11} - \eta^{10} - \\ &\qquad \eta^9 + \eta^8 + \eta^7 + \eta^6 + \eta^5 + \eta^4 - \eta^3 + \eta^2 + \eta + 1. \end{aligned}$$

Applying the algorithm in §4 above, one computes that the resultant of $p_1(w)$ and $p_2(w)$ is $h(S) = H(S^3 - S)$ where $H(t)$ is the product:

$$(\eta^{18} + \eta^{15} - \eta^{13} + \eta^{12} + \eta^{11} - \eta^{10} - \eta^9 - \eta^8 + \eta^6 + \eta^5 + \eta^4 - \eta + 1)*$$
$$(t - \eta^{17} - \eta^{14} + \eta^{13} - \eta^{12} - \eta^{11} + \eta^{10} + \eta^9 - \eta^7 - \eta^6 + \eta^5 - \eta^4 - \eta^3 - \eta^2)*$$
$$(t + \eta^{18} + \eta^{17} + \eta^{16} + \eta^{15} + \eta^{13} - \eta^{12} + \eta^{11} + \eta^{10} - \eta^9 + \eta^7 + \eta^5 - \eta^4 - \eta^3 + \eta + 1)*$$
$$(t^2 + (2\eta^{18} + \eta^{17} - \eta^{16} - \eta^{15} - \eta^{14} + \eta^{13} - \eta^{12} + \eta^{11} + \eta^{10} - \eta^9 - \eta^8 - \eta^6 - \eta^5 - \eta^4)t$$
$$+\eta^{18} + \eta^{17} - \eta^{16} + \eta^{15} + \eta^{13} - \eta^{11} + \eta^{10} + \eta^9 + \eta^7 + \eta^5 - \eta^2 + 1)*$$
$$(t^5 + (\eta^{17} - \eta^{15} - \eta^{14} - \eta^{12} - \eta^{11} + \eta^9 - \eta^7 + \eta^6 - \eta^5 + \eta^4 - \eta^3 - \eta^2 + 1)t^4$$
$$+(\eta^{18} + \eta^{17} + \eta^{16} + \eta^{15} - \eta^1 4 + \eta^8 - \eta^7 - \eta^3 + \eta^2 + \eta + 1)t^3$$
$$+(2\eta^{18} - \eta^{16} - \eta^{15} + \eta^{12} - \eta^{10} + \eta^9 - \eta^8 + \eta^6 + \eta^5 - \eta^4 + \eta - 1)t^2$$
$$+(2\eta^{16} - \eta^{15} - \eta^{13} + \eta^{11} + \eta^{10} - \eta^9 + \eta^8 - \eta^6 - \eta^5 - \eta^4 + \eta^3 + \eta^2 - \eta + 1)t$$
$$-\eta^{17} + \eta^{16} + \eta^{15} - \eta^{14} + \eta^{13} + \eta^{11} + \eta^{10} + \eta^9 + \eta^8 + \eta^7 - \eta^6 + \eta^3 - \eta^2 - \eta).$$

Let ρ_1 and ρ_2 be the two roots of $H(t)$ in $\mathbb{F}_q$ (in the order above). Then $S^3 - S - \rho_1$ is irreducible in $\mathbb{F}_{3^{19}}[S]$, but $S^3 - S - \rho_2 = (S - \delta)(S - \delta + 1)(S - \delta - 1)$ where

$$\delta = \eta^{18} + \eta^{17} + \eta^{15} - \eta^{14} + \eta^{12} + \eta^{11} + \eta^{10} - \eta^8 + \eta^7 - \eta^6 - \eta^5 - \eta^4.$$

All three of δ, $\delta + 1$, and $\delta - 1$ give s_0's such that $\sum_{i=0}^{4} s_i z^i$ is the x-coordinate of a point in the trace zero subgroup of $E_1(\mathbb{F}_{3^{95}})$.

6. $E(\mathbb{F}_{q^{2m}})$

In §2 we restricted to odd r. The case $r = 2m$ essentially reduces to the case $r = m$, as follows. In general, taking the Weil restriction of scalars, we have that $\mathrm{Res}_{M/k} E \sim E \oplus T$, where the trace zero variety T is the kernel of the trace map $\mathrm{Tr}_{M/k} : \mathrm{Res}_{M/k} E \to E$, and $\sim$ denotes a homomorphism with small kernel and cokernel. Letting $M = \mathbb{F}_{q^{2m}}$, $L = \mathbb{F}_{q^m}$, and $k = \mathbb{F}_q$, and letting E' denote the quadratic twist of E, one finds that

$$\mathrm{Res}_{M/k} E \cong \mathrm{Res}_{L/k} E \oplus \mathrm{Res}_{L/k} E'.$$

Dealing with the trace zero subgroup of $E(M/k)$ essentially reduces to dealing with $E'(L/k)$ and the trace zero subgroup of $E(L/k)$. As a practical matter, there does not seem to be any benefit in doing cryptography in the trace zero subgroup of $E(\mathbb{F}_{q^{2m}})$, rather than in $E(\mathbb{F}_{q^m})$ itself (or in $E'(\mathbb{F}_{q^m})$).

For example, when $r = 2$, then $\sigma(P) = -P$. Since the x-coordinates of points in the trace zero subgroup $A_0 \subset E(\mathbb{F}_{q^2})$ lie in $\mathbb{F}_q$, they are already compressed. Here, $A_0 \cong E'(\mathbb{F}_q)$.

7. Explanation of the algorithm

In §5.1 of [2] we explained why the algorithm works. We recall that explanation here (in greater generality).

Let X and Y denote the coordinate functions on the elliptic curve E. Since $\mathrm{Tr}(P) = O_E$, there is a function $\mathcal{F}(X, Y)$ on E with zeros at the points $\sigma^i(P)$ for $0 \le i \le r - 1$, a pole of order r at O_E, and no other zeros or poles. Writing $P = (s, t)$, then the function $g(X) = \prod_{i=0}^{r-1}(X - \sigma^i(s))$ can be viewed as a function on E with zeros at $\pm\sigma^i(P)$ for $0 \le i \le r - 1$, a pole of order $2r$ at O_E, and no other zeros or poles. Thus $g(X) = \gamma\mathcal{F}(X, Y)\tilde{\mathcal{F}}(X, Y)$, where $\tilde{\mathcal{F}}$ is $\mathcal{F}$ composed with

multiplication by -1 on E, and $\gamma \in \mathbb{F}_q^\times$. We can write $\mathcal{F}(X,Y) = h_1(X) + h_2(X)Y$ with $h_1(X), h_2(X) \in \mathbb{F}_q[X]$. Since X has a double pole at O_E and Y has a triple pole at O_E, it follows that $\deg(h_2) = (r-3)/2$ and $\deg(h_1) \leq (r-1)/2$ when r is odd, and $h_2(X) = 0$ and $h_1(X) = X - s$ when $r = 2$. We have $-(x,y) = (x, -y - a_1x - a_3)$ in E, so

$$\tilde{\mathcal{F}}(X,Y) = h_1(X) - h_2(X)(Y + a_1X + a_3)$$

and

$$\mathcal{F}(X,Y)\tilde{\mathcal{F}}(X,Y) = \\ h_1(X)^2 - h_1(X)h_2(X)(a_1X + a_3) - h_2(X)^2(X^3 + a_2X^2 + a_4X + a_6).$$

When r is odd, let $\gamma = -1$ and write

$$h_1(X) = \sum_{i=0}^{(r-1)/2} \alpha_i X^i, \quad h_2(X) = X^{(r-3)/2} + \sum_{i=0}^{(r-5)/2} \beta_i X^i$$

with the α_i's and β_i's in $\mathbb{F}_q$.

8. Security issues

Gaudry [1] recently announced a new attack on the discrete logarithm problem in $E(\mathbb{F}_{q^r})$ for small values of r. Theorem 1 of [1] states that (using improvements due to Thériault) one can obtain an algorithm that runs in time $O(q^{2-(4/(2r+1))})$ (up to logarithmic factors). In the case of $E(\mathbb{F}_{q^5})$, this gives complexity $O(q^{18/11}\log(q)^k)$ for some k. Since the trace zero subgroup A_0 has size about q^4, the Pollard rho algorithm has complexity $O(q^2)$, which seems to be a weaker attack, except possibly for the fact that the constant (and possibly the logarithmic factor) in the Gaudry/Thériault attack can be quite large. In the case of $E(\mathbb{F}_{q^3})$, the subgroup A_0 has size about q^2, so the ($O(q)$) Pollard rho attack is stronger than the Gaudry/Thériault attack.

References

[1] P. Gaudry, *Index calculus for abelian varieties and the elliptic curve discrete logarithm problem*, Cryptology ePrint Archive, Report 2004/073, `http://eprint.iacr.org/2004/073` .

[2] K. Rubin, A. Silverberg, *Supersingular abelian varieties in cryptology*, in Advances in Cryptology — CRYPTO 2002, Lect. Notes in Comp. Sci. **2442**, Springer, Berlin, 2002, 336–353.

[3] K. Rubin, A. Silverberg, *Using primitive subgroups to do more with fewer bits*, in Algorithmic Number Theory (ANTS VI), Lect. Notes in Comp. Sci. **3076**, Springer, Berlin, 2004, 18–41.

Department of Mathematics, University of California at Irvine, Irvine, CA 92697-3875, USA

E-mail address: `asilverb@uci.edu`