Algebraic Cryptography
Homework 6

Due Monday, 4 December 2017

Problem 1. Show that a linear change of variables can be used to transform the left side of the equation

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

over a field $\mathbb{F}$ to the form:

(a) y^2 if $\mathrm{Char}(\mathbb{F}) \neq 2$;

(b) $y^2 + xy$ if $\mathrm{Char}(\mathbb{F}) = 2$ and the xy-term in equation (1) is nonzero (i.e., $a_1 \neq 0$).

Proof. (a) We will make the substitution $y = y' + bx + c$. Then the terms on the left-hand side become

$$\begin{aligned} y^2 &= y'^2 + 2bxy' + 2cy' + (2bcx + b^2x^2 + c^2) \\ a_1xy &= a_1xy' + (a_1bx^2 + a_1cx) \\ a_3y &= a_3y' + (a_3bx + a_3c). \end{aligned}$$

All the terms involving only constants and factors of x (terms in parentheses above) will be moved to the right-hand side and so can be ignored. Then after the substitution and rearrangement, the left-hand side becomes

$$y'^2 + (2b + a_1)xy' + (2c + a_3)y',$$

and so to make the coefficients of the latter terms zero, we let $b = \frac{-a_1}{2}$ and $c = \frac{-a_3}{2}$ (we can divide by 2 since $\mathrm{Char}(\mathbb{F}) \neq 2$).

(b) One can check that making a substitution of the form $y = y' + bx + c$ doesn't change the left-hand side at all when $\mathrm{Char}(\mathbb{F}) = 2$. So we should consider a substitution of x. To keep things simple, we will only consider a substitution of the form $x = x' + b$ (because including a y term in this substitution would cause some terms from the right-hand side to move to the left, which would make things hard to keep track of; a substitution

like we have chosen isn't guaranteed to work, but hopefully it will). Under our substitution, the left-hand side becomes

$$y^2 + a_1xy + a_3y = y^2 + a_1x'y + (a_1b + a_3)y$$

and so setting $b = \frac{-a_3}{a_1}$ (we can divide by a_1 since it is nonzero), we can force the coefficient of y to be zero. This leaves us with an equation whose left-hand side has the form $y^2 + a_1x'y$. In order to make the a_1 coefficient 1, we perform the substitutions $y = a_1^3y''$ and $x' = a_1^2x''$. This yields an equation in which the y^2, xy, and x^3 terms all have the same coefficient (a_1^6), and so we can divide by it to make all of them have a coefficient of 1, as desired.

■

Problem 2. If $\text{Char}(\mathbb{F}) = 2$, show that there is no elliptic curve with equation (1) where $a_1 = a_3 = 0$.

Proof. Recall that an equation of the form (1) defines an elliptic curve over $\mathbb{F}$ only if it is *smooth*; that is, its partial derivatives have no simultaneous solutions in $E(\overline{\mathbb{F}})$. If $\mathbb{F}$ has characteristic 2 the partial derivative with respect to x yields the equation:

$$\begin{aligned} a_1y &= 3x^2 + 2a_2x + a_4 \\ a_1y &= 3x^2 + a_4. \end{aligned}$$

The partial derivative with respect to y yields the equation:

$$\begin{aligned} 2y + a_1x + a_3 &= 0 \\ a_1x + a_3 &= 0. \end{aligned}$$

If $a_1 = a_3 = 0$, then these two equations become

$$\begin{aligned} 0 &= 3x^2 + a_4 \\ 0 &= 0 \end{aligned}$$

which obviously have simultaneous solutions in the algebraic closure of $\mathbb{F}$, namely the pair $(\sqrt{-a_4/3}, y)$ for any value of y. To conclude the curve is not smooth, we must check that one of these points is on the elliptic curve. That is, we need to check that it satisfies the equation $y^2 = x^3 + a_2x^2 + a_4x^4 + a_6$. However, since we have a free choice of y, this is trivial. Thus, when $a_1 = a_3 = 0$, the curve defined by (1) is not smooth and therefore not an elliptic curve. ■

Problem 3. In the case when $\text{Char}(\mathbb{F}) \neq 2$, from the first problem we can write an equation for an elliptic curve in the form

$$y^2 = x^3 + a_2x^2 + a_4x + a_6, \tag{2}$$

where the coefficients on the right are possibly different from the original ones, but this is not important. Show that this curve is smooth if and only if the cubic polynomial on the right has no multiple roots (in the algebraic closure $\overline{\mathbb{F}}$).

Proof. As in the previous problem, we take the partial derivatives with respect to x and y and end up with the equations

$$\begin{aligned} 0 &= 3x^2 + 2a_2x^2 + a_4x + a_6 \\ 2y &= 0. \end{aligned}$$

Notice that if the cubic on the right-hand side of (2) has a multiple root $\alpha \in \overline{\mathbb{F}}$, then it is also a root of its partial derivative, and so satisfies the first equation in the system above as well. Therefore, the pair $(\alpha, 0) \in E(\overline{\mathbb{F}})$ and also simultaneously satisfies both partial derivative equations. Thus the curve is not smooth in this case.

Conversely, suppose that the curve is not smooth; that is, suppose that there is a point $(\alpha, \beta) \in E(\overline{\mathbb{F}})$ which satisfies both partial derivative equations. Then, since it satisfies the second, that implies $\beta = 0$. Because $(\alpha, 0) \in E(\overline{\mathbb{F}})$ and satisfies the first partial derivative equation, we see that α is a root of both the cubic and its derivative. Therefore the cubic has α has a multiple root. ■

Problem 4. Each of the following points has finite order on the given elliptic curve over $\mathbb{Q}$. In each case, find the order of P.

(a) $P = (0, 16)$ on $y^2 = x^3 + 256$.

(b) $P = \left(\frac{1}{2}, \frac{1}{2}\right)$ on $y^2 = x^3 + \frac{1}{4}x$.

(c) $P = (3, 8)$ on $y^2 = x^3 - 43x + 166$.

Proof. We will just compute a few multiples of P in each case.

(a) For $P = (0, 16)$ on $y^2 = x^3 + 256$, note that $2P = (0, -16)$, and thus $3P = 2P + P = O$. Thus the order of P is 3.

(b) For $P = \left(\frac{1}{2}, \frac{1}{2}\right)$ on $y^2 = x^3 + \frac{1}{4}x$, note that $2P = (0, 0)$ and hence $4P = O$. Thus the order of P divides 4, but it is not 2, and hence it must be 4.

(c) For $P = (3, 8)$ on $y^2 = x^3 - 43x + 166$, note that $2P = (-5, -16)$, $4P = (11, 32)$, and $8P = (3, 8) = P$. Therefore $7P = O$, and since 7 is prime, the order of P must be 7.

■

Problem 5. (a) Describe in detail the Elliptic Curve Diffie–Hellman Key Exchange and ElGamal Message Transmission methods.

(b) Consider the Elliptic Curve Diffie–Hellman Problem (ECDHP), the ElGamal Problem (EGP) and the Elliptic Curve Discrete Log Problem (ECDLP). Prove that ECDHP and EGP are equivalent, and that both problems reduce to ECDLP.

(c) Explain why (we think) that ECDHP and EGP are secure schemes.

Proof. In what follows, we consider an elliptic curve E over a finite field $\mathbb{F}$, which are assumed to be publicly known.

(a) The Elliptic Curve Diffie–Hellman Key Exchange protocol has the following steps:

   (a) Alice and Bob publicly agree on a base point P on E.
   (b) Alice picks a random integer a and Bob picks a random integer b (each choice is private).
   (c) Alice sends Bob (publicly) aP and Bob sends Alice (publicly) bP.
   (d) Alice computes $a(bP)$ and Bob computes $b(aP)$, which are both the shared secret key $(ab)P$.

   The ElGamal Message Transmission protocol has the following steps.

   (a) Alice and Bob publicly agree on a base point P on E.
   (b) Alice picks a random integer a and Bob picks a random integer b (each choice is private).
   (c) Alice sends Bob (publicly) aP and Bob sends Alice (publicly) bP.
   (d) Alice picks another random integer k.
   (e) To send a message m, Alice sends Bob the pair of points kP and $m + k(bP)$.
   (f) To decrypt, Bob computes $b(kP) = k(bP)$ and then computes $(m + k(bP)) - k(bP) = m$.

(b) For clarity, we state each of the problems:

   ECDHP Input: P, aP, bP. Output: $(ab)P$.

   EGP Input: $P, aP, bP, kP, m + k(bP)$. Output: m.

   ECDLP Input: P, aP. Output: a.

   We first prove that EGP reduces to ECDHP. Indeed, let $P, aP, bP, kP, m + k(bP)$ be an instance of EGP. Then P, bP, kP is an instance of ECDHP, and so by calling ECDHP, we obtain $(bk)P = k(bP)$. Subtracting this from $m + k(bP)$ we obtain m.

   We now prove that ECDHP reduces to EGP. Indeed, let P, aP, bP be an instance of ECDHP. Then $P, 0P, bP, aP, O$ is an instance of EGP (since $O = -a(bP) + a(bP)$), and so by applying EGP, we obtain $-a(bP)$. Negating this (which is a trivial operation, just negate the second coordinate), we obtain $a(bP) = (ab)P$.

   The previous paragraphs show that ECDHP and EGP are equivalent problems. Therefore, to show that they both reduce to ECDLP, it suffices to show that ECDHP reduces to ECDLP. For this, let P, aP, bP be an instance of ECDHP. Applying ECDLP to the instance P, aP, we obtain a. Then we compute$a(bP) = (ab)P$.

(c) We *think* that ECDHP and EGP are secure schemes because we have not found any more efficient way to solve them than by first solving ECDLP. Moreover, we have not found any polynomial time algorithms for ECDLP. Finally, elliptic curve cryptography (and these schemes) have been around about 30 years, so a trivial attack that no one has thought of yet seems unlikely.

■

Problem 6. Let G be a group whose order is B-smooth. Prove that there is a polynomial time algorithm (the input size is B) to solve the Discrete Log Problem to base $g \in G$. More specifically, given $g, y \in G$, your algorithm should output an integer x less or equal to the order of g such that $g^x = y$; *or state that no such x exists.* (Note: for this problem you may assume without proof that the Chinese Remainder Theorem has an $O(\ln(N)^2)$ algorithm where N denotes the "big" modulus in the CRT, i.e., $N = n_1 \cdots n_k$ where $n_1, \ldots, n_k$ are relatively prime).

Proof. Consider a group G of B-smooth order and elements $g, y \in G$. Our goal is to find an integer x (with $x \leq o(g)$) such that $xg = y$. Let $|G| = \prod_{i=1}^{l} p_i^{s_i}$ be the prime factorization. We can obtain any prime factor of $|G|$ in at most $O(B)$ divisions (using simple trial division) of $|G|$ by a number less than B. Each division takes a number of bit operations which is a polynomial in $\log_2 |G|$ and $\log_2 |B|$. Thus, the entire factorization is polynomial time in $\log_2 |G|$ and B. Note that $p_i \leq B$ and $l \leq B$ (the latter because there are not more than B distinct primes less than B). Moreover, note that $\sum_{i=1}^{l} s_i \leq \log_2 |G|$ because $|G| \geq \prod_{i=1}^{l} 2^{s_i} = 2^{\sum_{i=1}^{l} s_i}$.

Our first step will be to compute the order $o(g)$. Note that $o(g) \mid |G|$, so our goal will be to find the smallest power of each p_i such that $\left(\frac{|G|}{p_i^{r_i}}\right) g = O$. For this compute the sequence $\left(\frac{|G|}{p_i^{j}}\right) g$ for $j = 1, \ldots, s_i$, stopping at the first value of j for which this group element is not the identity. For this value of j, we find $r_i = s_i - j + 1$. Note, for each i, this takes at most s_i divisions and s_i groups operations. Thus, the whole procedure takes at most $\sum_{i=1}^{l} s_i \leq \log_2 |G|$ divisions and group operations, each of which can be performed in polynomial time. Thus this entire paragraph can be performed in polynomial time with input size $\log_2 |G|$. Finally, $N := o(g) = \prod_{i=1}^{l} p_i^{r_i}$.

Our next step will be, given $p^r := p_i^{r_i}$ for some i, to compute that base-p expansion of any $x \pmod{p^r}$. In particular, $x \equiv x_0 + x_1 p + \cdots + x_{r-1} p^{r-1} \pmod{p^r}$, with $x_j \in \{0, 1, \ldots, p-1\}$, and our goal is to compute all the x_j. We start by computing x_0. First notice that if $xg = y$ (which is what we are trying to do), then $\frac{Nx}{p} g = \frac{N}{p} y$. We can compute the right-hand side because we have that information, but notice that the left-hand side becomes $\frac{Nx_0}{p} g$, which happens because when we multiply by $\frac{N}{p}$, all the other terms in the expansion of x become multiples of N, which is the order of g. So, to find x_0, we do so by trial-and-error. Namely, we compute the sequence $\frac{N}{p} g, \frac{2N}{p} g, \ldots, \frac{(p-1)N}{p} g$ (this is

at most B group operations) until it matches $\frac{N}{p}y$, and this is our value of x_0. To find x_1, we first note that $(x-x_0)g = y - x_0 g$. Multiplying both sides by $\frac{N}{p^2}$, we obtain $\frac{Nx_1}{p^2} = \frac{N}{p^2}(y - x_0 g)$. Then, we perform the same trial-and-error procedure as before to find x_1. This also takes B groups operations. Repeating this for each $0 \leq j \leq r-1$ we find that there were about rB group operations (as well as r divisions). Thus, it takes $r_i B$ groups operations and r_i divisions to compute $x \mod p_i^{r_i}$. Doing this for each $1 \leq i \leq l$ takes $B \sum_{i=1}^{l} r_i \leq B \log_2 |G|$ group operations and not more than $\log_2 |G|$ divisions. Thus, computing $x \mod p_i^{r_i}$ for each $1 \leq i \leq l$ is polynomial time in the inputs $B, \log_2 |G|$.

Finally, we apply the Chinese Remainder Theorem to $x \pmod{p_i^{r_i}}$ to find $x \pmod{\prod_{i=1}^{l} p_i^{r_i}} = x \pmod{N}$. This is polynomial time in the logarithms of $\max p_i^{r_i}$ and l, so polynomial time in $\log_2 p_i^{r_i} \leq r_i \log_2 p_i \leq \log_2 |G| \log_2 B$ and $\log_2 l \leq \log_2 B$. This step is much faster than the previous paragraph.

This entire algorithm computes x (if it exists). If not such x exists, then at some point there will be no match at the trial-and-error step. If that happens, output "no such x exists". ■