

## Takehome Project 2 Written Solutions

## Written Part

7. Let's begin the written portion of this assignment by studying the correctness of the MV-Elgamal.

- (a) Let  $E$  be an elliptic curve modulo a prime  $p$  and  $P \in E(\mathbb{F}_p)$  be a point not equal to  $\mathcal{O}$ . Prove that the order of  $P$  is 2 if and only if its  $y$  coordinate is equal to 0. (This justifies the remark in 5(a)).

*Proof.* Let  $P = (x, y)$ . Since  $P \neq \mathcal{O}$  we may assume we get to step 3 of the algorithm, and observe that  $P + P = \mathcal{O}$  if and only if  $x = x$  and  $y \equiv -y \pmod{p}$ . This occurs precisely when  $y \equiv 0 \pmod{p}$  (notice here we are using that  $p \neq 2$  in an important way, as  $1 \equiv -1 \pmod{2}$ ).  $\square$

- (b) Use part (a) to explain why in the key creation phase of MV-Elgamal it was insisted that  $Q_A$  must have a nonzero  $y$  coordinate.

*Proof.* When encrypting a message we assume the  $x$  and  $y$  coordinates of  $S$  are nonzero—this is because we must be able to divide by them to recover the message, as explained in part (c). But  $S$  is a multiple of  $Q_A$ . If  $Q_A$  has 0 for its  $y$ -coordinate, then  $Q_A$  has order 2, so that its only multiples are  $\mathcal{O}$  and itself. But  $S$  can't be either of these points, because in the first case there are no  $x$  and  $y$  coordinates, and in the second, we would have  $y_S = 0$ . In either case we would not be able to encrypt a message with  $S$ .  $\square$

- (c) Prove the correctness of MV-Elgamal. (Be sure to explain why  $x_S, y_S$  must be nonzero).

*Proof.* We first observe that:

$$T = n_A R = n_A(kP) = k(n_A P) = kQ_A = S.$$

Therefore  $(x_T, y_T) = (x_S, y_S)$ . As we ensured that  $x_S$  and  $y_S$  are nonzero (modulo  $p$ ), we see that so are  $x_T$  and  $y_T$ , and therefore it is in fact possible to compute  $x_T^{-1}$  and  $y_T^{-1}$ . The identity  $x_T^{-1} x_S = 1$  is immediate, as is  $y_T^{-1} y_S = 1$ . Therefore we observe that

$$m'_1 = x_T^{-1} c_1 = x_T^{-1} x_S m_1 = m_1 \quad \text{and} \quad m'_2 = y_T^{-1} c_2 = y_T^{-1} y_S m_2 = m_2.$$

$\square$

- (d) The original Elgamal and the Elliptic Curve Elgamal essentially wrap together a Diffie-Hellman key exchange and a symmetric cipher into one procedure. Explain why this is also the central philosophy of MV-Elgamal. What plays the roll of the shared secret? What about the symmetric key?

*Proof.* The Diffie-Hellman component of MV-Elgamal can be summarized as follows: Alice has a secret multiplier  $n_A$ , while Bob has  $k$ . Alice multiplies  $P$  by her secret multiplier, and sends it to Bob, who does the same to arrive at  $S$ . Conversely, Bob

multiplies  $P$  by his secret multiplier and sends it to Alice, who then multiplies by hers to arrive at  $T$ . As we saw above,  $T = S = kn_AP$  is the shared secret.

This shared secret is used to construct a symmetric cipher as follows: The coordinates of  $S$  (or  $T$ ) are invertible elements  $x_S, y_S \in \mathbb{F}_p^*$ , that Alice and Bob both secretly know. They use these values as symmetric keys for the *multiplication mod  $p$*  cipher. That is, Bob multiplies his message(s) by the coordinate(s), to produce the ciphertext, and Alice divides the ciphertext(s) by the coordinate(s) to recover the plaintext.  $\square$

## 8. Let's now discuss the efficiency of MV-Elgamal

- (a) What is the message expansion of MV-Elgamal?

*Proof.* The message is a pair  $(m_1, m_2)$  with  $m_i \in \mathbb{F}_p^*$ . If  $p$  is approximately  $k$ -bits in size, then this means we can send messages up to  $2k$ -bits long. To send a message one must send a point on  $E(\mathbb{F}_p)$  ( $R$ ) and two elements of  $\mathbb{F}_p^*$  ( $c_1$  and  $c_2$ ). In particular, this is 4 numbers  $k$ -bits long. Therefore it takes a cipher approximately  $4k$  bits long to send approximately  $2k$  bits of data, resulting in a 2-1 message expansion.  $\square$

- (b) Suppose Alice has an algorithm to efficiently compute square roots modulo  $p$  (such things certainly exist). Explain a way that Bob could compress the ciphertext by sending 1 bit in place of the  $y$ -coordinate of  $R$ , and prove that your method works. What would the new message expansion be?

*Proof.* Notice that if  $R = (x, y)$  then the coordinates  $x$  and  $y$  satisfy the equation  $y^2 = x^3 + Ax + B$  (where  $A$  and  $B$  are public). Therefore once you know  $x$ , you know the value of  $y^2$ . Assuming you know how to compute square roots modulo  $p$ , then you can narrow the value of  $y$  down to two values, the two square roots of  $y^2$  (HW2 Problem 8). We know that these are  $y$  and  $p - y$  modulo  $p$ . Notice that if  $y < p/2$  then  $p - y > p - p/2 = p/2$ . In particular, exactly one of the square roots is less than  $p/2$ , and the other is greater.

To exploit this, Bob instead of sending the point  $R = (x, y)$ , can send the pair  $(x, b)$  where  $b \in \{0, 1\}$  is a bit. Then the agreed upon rule could be:

$$b = \begin{cases} 0 & 0 \leq y < p/2 \\ 1 & p/2 \leq y < p \end{cases}$$

Then Alice can compute  $y$  and  $p - y$  from  $x$  as described in the previous paragraph, and deduce which one is  $y$  given the value of  $b$ .

(Alternatively, as  $p$  is odd, one can observe that  $y$  is even if and only if  $p - y$  is odd, and the extra bit can encode whether the  $y$ -coordinate is even or odd.)

(Alternatively, the extra bit could encode whether the  $y$ -coordinate is the larger or smaller square root of  $y^2$ .)

Since we don't have to send the  $y$ -coordinate, Bob only needs to send three elements of  $\mathbb{F}_p$  (plus one bit) as the ciphertext. In particular, it takes  $3k + 1$  bits to encode a message  $2k$  bits long, so that the message expansion becomes 1.5-1 (plus one extra bit for  $b$ ).  $\square$

## 9. Let's conclude with some potential attacks on MV-Elgamal.

- (a) Show that if Eve can solve the Elliptic Curve Discrete Log Problem she can use it to crack MV-Elgamal.

*Proof.* Eve can consult an ECDLP oracle, presenting the oracle with  $P$  and  $n_AP = Q_A$  in  $E(\mathbb{F}_p)$ . The oracle will then present Eve with the value  $n_A$ . This is all Eve needs to crack MV-Elgamal. Indeed, with this in hand, if Eve intercepts Bob's message  $(R, c_1, c_2)$ , where  $R = kP$  and  $(c_1, c_2) = (x_S m_1, y_S m_2)$  where  $S = kQ_A$ . Then Eve can compute  $n_AR = n_A kP = kQ_A = S$ , can compute  $x_S^{-1}$  and  $y_S^{-1}$ , and multiply them by  $c_1$  and  $c_2$  respectively to recover the message.  $\square$

- (b) Eve knows the elliptic curve  $E$ , and intercepted the two peices of ciphertext  $c_1, c_2$ . Explain how Eve could use this information to relate  $m_1$  and  $m_2$  in a polynomial equation (modulo  $p$ ). In particular, if Eve knows one of the peices of plaintext, she can use that to recover the other peice by solving an equation modulo  $p$ . (The moral here is, both peices of plain text must be unique and private with each message. For example: Bob should never leave one blank just because he doesn't need the space.)

*Proof.* We have relations  $c_1 = x m_1$  and  $c_2 = y m_2$  for some point  $(x, y)$  on a publicly known elliptic curve. In particular, since  $c_i, m_i \in \mathbb{F}_p^*$  we can deduce that  $(x, y) = (c_1/m_1, c_2/m_2)$  (where here division is done modulo  $p$ ). Therefore the following equation (or rather congruence modulo  $p$ ) holds:

$$\left(\frac{c_2}{m_2}\right)^2 = \left(\frac{c_1}{m_1}\right)^3 + A\left(\frac{c_1}{m_1}\right) + B.$$

Therefore, if we know intercept the ciphertext, and know  $m_1$ , we can use this equation to compute  $\left(\frac{c_2}{m_2}\right)^2$ , and therefore taking square roots get two possible values for  $c_2/m_2$ . This allows us to solve for  $m_2$  narrowing down the search for the plaintext to two values. Conversely, knowing  $m_2$  we would have  $x = \frac{c_1}{m_1}$  in a cubic equation (mod  $p$ ). The cubic formula therefore reduces this computation to taking cube roots and square roots, and gives 3 possible values for  $x$ , and therefore for  $m_1$ .  $\square$

- (c) Connect 9(b) to the structure of the messages we exchanged in the communication part of this assignment.

*Proof.* When I sent my message I put one of your secret numbers in  $m_1$  and a different one in  $m_2$ , and you responded with their sum in  $m_1$  and their product in  $m_2$ . This guaranteed that  $m_1$  and  $m_2$  were unique to each of you, and that no one had enough information to find just one. For example, if I had sent the numbers in the first message, and told you what to do in the second, I would have been sending the same second message to everybody in the class. This could have allowed folks to use that information and the techniques described in part (b) to deduce  $m_1$  and therefore get your secret number.  $\square$