MATH 4440/5440 Assessment, Module 4 (Finite fields, elliptic curve cryptography, lattice-based cryptography)

Katherine Stange, CU Boulder, Fall 2020

Honor Code Rules

Assessments are open book, but are to be completed on your own without collaboration. To be specific, you may use your course notes, textbook, course website resources, course videos. You may not use the internet beyond the course websites. You may not ask anyone else for help (except your professor), including other humans, or posting/entering your question or any part of it into the internet. You may not share the questions or answers with anyone else. You may not use calculators (even from the course websites) unless explicitly permitted in the question.

Have you read, understood, and followed the honor code rules above?

$$YES \quad / \quad NO$$

Some instructions on formatting.

You may use the accompanying LaTeX source document to produce LaTeX'ed answers. You may typeset answers separately. You may print the pages and solve the questions on them by hand. You may handwrite answers on separate sheets. You may upload PDF or image files (JPG or PNG). No matter what you do, just make sure it is clearly and easily legible before you upload it to canvas.

1 Question 1

(10 points) Determine the inverse of $X^2 + 1$ in the finite field $\mathbb{F}_{27}$ determined by the irreducible polynomial $X^3 - X + 1$.

You may not use a calculator/computer. Please show your work neatly and **show all steps**. When you are done, double-check your work.

Solution.

We can use the extended Euclidean algorithm to find this inverse. We need to solve

$$s(X^3 - X + 1) + t(X^2 + 1) = 1.$$

The regular Euclidean algorithm would look like this:

$$\begin{aligned} X^3 - X + 1 &= X(X^2 + 1) + (X + 1) \\ X^2 + 1 &= (X + 2)(X + 1) + 2 \\ X + 1 &= (2X + 2)2 + 0 \end{aligned}$$

There aren't too many steps here, so you could back-substitute or other method to write 2 as a linear combination of $X^2 + 1$ and $X^3 - X + 1$. But the extended Euclidean algorithm method would look like this (where my 'cards' look like triple):

$$\begin{aligned} (X^3 - X + 1, s = 1, t = 0) &= X(X^2 + 1, s = 0, t = 1) + (X + 1, s = 1, t = 2X) \\ (X^2 + 1, s = 0, t = 1) &= (X + 2)(X + 1, s = 1, t = 2X) + (2, s = 2X + 1, t = X(X + 2) + 1) \end{aligned}$$

That tells us that

$$(2X + 1)(X^3 - X + 1) + (X(X + 2) + 1)(X^2 + 1) = 2$$

At this point we should quickly verify this is accurate:

$$(2X+1)(X^3-X+1)+(X(X+2)+1)(X^2+1) = 2X^4+X^2+2X+X^3+2X+1+X^4+2X^3+X^2+X^2+2X+1 = 2$$

So we have found that

$$(X^2 + 2X + 1)(X^2 + 1) \equiv 2 \pmod{X^3 - X + 1}$$

This isn't quite what we need (it comes out to 2 instead of 1) but that is easily remedied:

$$(2X^2 + X + 2)(X^2 + 1) \equiv 1 \pmod{X^3 - X + 1}$$

So the inverse is $2X^2 + X + 2$.

2 Question 2

(10 points) Consider the elliptic curve $y^2 = x^3 + x + 3$ over the finite field $\mathbb{F}_7$. Double the point $P = (4, 1)$.

You may not use a calculator/computer. Please show your work neatly and **show all steps**. When you are done, double-check your work.

Solutions

To double this point, we need to find the tangent line at $P = (4, 1)$. To do this, we take the derivative of the curve equation

$$2y\frac{dy}{dx} = 3x^2 + 1$$

Solving this gives

$$\frac{dy}{dx} = \frac{3x^2 + 1}{2y} = \frac{3(4^2) + 1}{2} = \frac{0}{2} = 0.$$

So the slope at P is 0. The line equation is therefore $y = 1$.

Next, we find the third intersection point of this line with the curve. We have

$$(1)^2 = x^3 + x + 3.$$

This becomes

$$x^3 + x + 2 = 0.$$

The roots of this sum to the negative of the coefficient of x^2, i.e. 0. Two of the roots are 4 and the third is the one we seek, say x_R:

$$4 + 4 + x_R = 0.$$

We solve and obtain

$$x_R = 6.$$

The corresponding y_R satisfies the line equation, i.e. $y_R = 1$. So we get

$$R = (6, 1).$$

Reflecting this across the x-axis (negating) gives

$$2P = (6, 6).$$

3 Question 3

(10 points)

1. Count the number of points on the projective plane $\mathbb{P}^2$ over the finite field $\mathbb{F}_q$ (for example, in class we saw that for $\mathbb{F}_3$, there were 13 points).

2. Define projective n-space over $\mathbb{F}_q$ as follows (this generalizes what we did in class):

$$\mathbb{P}^n = \{\mathbf{v} \in \mathbb{F}_q^{n+1} \setminus \{0\}\}/\sim$$

where the $\sim$ is an equivalence relation as follows: $\mathbf{v} \sim \mathbf{w}$ iff $\mathbf{v} = \lambda\mathbf{w}$ for some $\lambda \in \mathbb{F}_q^*$. How many elements are there in $\mathbb{P}^n$ over $\mathbb{F}_q$?

Solution

We will solve the second part first, since the first part is just a special case of the second.

There are q^{n+1} elements of $\mathbb{F}_q^{n+1}$, since these are vectors whose $n+1$ entries can each take q possible values. There are $q^{n+1} - 1$ non-zero such vectors.

Under the equivalence relation, exactly $q - 1$ vectors are identified in one equivalence class (since there are $q - 1$ non-zero values for λ in the definition above).

Therefore the number of equivalence classes is

$$\frac{q^{n+1} - 1}{q - 1} = q^n + q^{n-1} + \cdots + q + 1.$$

4 Question 4

(10 points) Find the shortest lattice vector in the lattice spanned by $(17, 5)$ and $(27, 13)$.

You may not use a calculator/computer **except for adding/subtracting/multiplying integers**. Please show your work neatly and **show all steps**. When you are done, double-check your work.

Solution We use the lattice reduction method.

Let $v = (17, 5)$ and $w = (27, 13)$. These satisfy

$$v \cdot v = 314, w \cdot w = 898, v \cdot w = 524.$$

Therefore w is the longer, so we compute

$$\frac{v \cdot w}{v \cdot v} = \frac{524}{314}.$$

The closest integer here is 2, so our new basis is $v \leftarrow (17, 5)$ and $w \leftarrow w - 2v = (-7, 3)$. These satisfy

$$v \cdot v = 314, w \cdot w = 58, v \cdot w = -104.$$

Therefore v is the longer, so we compute

$$\frac{v \cdot w}{w \cdot w} = \frac{-103}{58}.$$

The closest integer here is -2, so our new basis is $w \leftarrow w = (-7, 3)$ and $v \leftarrow v + 2w = (3, 11)$. These satisfy

$$v \cdot v = 130, w \cdot w = 58, v \cdot w = 12.$$

Therefore v is the longer, so we compute

$$\frac{v \cdot w}{w \cdot w} = \frac{12}{58}.$$

The nearest integer is 0, which means that we are done, and w must be the shortest vector: $(-7, 3)$.

5 Question 5

(10 points) In class, I stated but then did not prove that Ring-LWE security depends on the Decision Ring-LWE problem. In this problem, I ask you to essentially give the idea of that proof.

Ring-LWE Setup

We use the standard notation of Ring-LWE encryption as in lecture. In particular, the ring R_p is decided (including n and p), as is k.

Definition of a collection of samples being "Ring-LWE"

If $(a_1, b_1), \ldots, (a_n, b_n) \in R_p \times R_p$, then we say that this collection of samples is a "Ring-LWE" collection if there is a short $s \in R_p$ such that they all have the form $(a_i, b_i = a_i s + e_i)$ for short e_i. Otherwise we say they are not Ring-LWE samples.

Probability Assumptions

1. Suppose that the probability that a *randomly* generated collection of *one* sample is Ring-LWE is 1/2. (By randomly I mean the entries of the sample are uniformly randomly chosen from R_p.)
2. Suppose that the probability that a *randomly* generated collection of *two or more* samples is Ring-LWE is extremely small (so that you can expect it won't occur at all during your experiments).

Definition of a well-formed ciphertext

The data of a purported public key (a, b), ciphertext (v, w) and plaintext m is "well-formed" if both of the following hold:

1. (a, b) is a valid public key (i.e. is of the form $(a, b = as + e)$ for some s and e which are short), and
2. (v, w) is a possible valid Ring-LWE ciphertext for the plaintext m encrypted to the public key (a, b).

The Oracle

Suppose that you have access to a black box machine (an "Oracle") that can discern whether a purported Ring-LWE public key (a, b) and ciphertext (v, w) is a well-formed ciphertext for the purported message m. In other words, if it is given (a, b), (v, w) and m, it will return YES or NO, completely reliably (i.e. it is always correct).

The Challenge

Suppose that you are given two samples (a_1, b_1) and (a_2, b_2). You know that these samples are either randomly generated (with probability 1/2) or Ring-LWE (with probability 1/2). You wish to determine if this pair of samples is a Ring-LWE collection or not (see definition above). The size of the ring R_p rules out any hope of an exhaustive search or other exponential attack.

How can you use the Oracle to make a better than random guess? Explain why it works, and what the probability of guessing correctly is.

Hint: The trick here is to somehow create a fake "challenge" (a, b), (v, w), m for the Oracle and then use its response to help you make your guess.

Solution # 1: 3/4 chance of being correct with one call to oracle

Algorithm

The idea is to choose $m = 0$, set $(a, b) = (a_1, a_2)$ and $(v, w) = (b_1, b_2)$. (It's also possible to do a version where you pick m to be some other random thing.) Return YES when the oracle returns YES and NO when the oracle returns NO.

Analysis

There's a probability 1/2 that (a, b) is a valid public key. If it is not, the Oracle will return NO. If it is, the Oracle will detect whether $b_1 = a_1 r + e_1$ and $b_2 = a_2 r + e_2$ for the same short r and return YES if so, and NO if not.

To analyse when you are correct:

1. If (a_1, a_2) is a valid public key (probability 1/2), then you will answer correctly all the time.
2. If (a_1, a_2) is not (probability 1/2), then you will answer NO all the time (and be correct 1/2 the time).

Therefore, you have a 3/4 chance of being correct.

Solution # 2: 7/8 chance with two calls to oracle

Algorithm

1. First, use $(a, b) = (a_1, b_1)$ as the public key and use a random ephemeral key to encrypt a random message m as (v, w). Send this to the oracle, which gives us response R_1 (either True or False).
2. Then, use $(a, b) = (a_2, b_2)$ as the public key and obtain response R_2.

Based on the results, if the answers are all True, then return True to the challenge. Otherwise, return False.

Analysis

If the samples are random, then R_1 will be randomly T or F with equal probability (by the probability assumptions). Similarly for R_2. Therefore there's a 1/4 probability of getting all True, and causing you to err in your response to the challenge.

If the samples are Ring-LWE, then both will return True and you will answer correctly.

Therefore, on a random challenge, there's a 1/8 probability of error in your response. In other words, you have probability 7/8 of being correct.

Solution # 3 (based on a solution of J. Cates)

The following increases your probability of success and deals with the issue of independence of the samples.

Algorithm

1. First, use $(a, b) = (a_1, b_1)$ as the public key and use a random ephemeral key to encrypt a random message m as (v, w). Send this to the oracle, which gives us response R_1 (either True or False).
2. Then, use $(a, b) = (a_2, b_2)$ as the public key and obtain response R_2.
3. Then, use $(a, b) = (a_1 + a_2, b_1 + b_2)$ as the public key and obtain response R_3.

Based on the results, if the answers are all True, then return True to the challenge. Otherwise, return False.

Analysis

If the samples are random, then R_1 will be randomly T or F with equal probability (by the probability assumptions). Similarly for R_2 and for R_3. The question is whether there is any correlation between the answers: pairwise the answer is no, since they differ by a random difference. However, it may be possible for them to have a correlation between all three; this is not clear from the information in the problem. It is unlikely, however. Therefore there's a 1/8 probability of getting all True.

If the samples are Ring-LWE, then all three will return True. Note that the third will return True because the two samples *share the same secret* so that the sum of the samples has the same secret also, i.e. if $(a_i, b_i) = (a_i, a_i s + e_i)$ then $(a_1 + a_2, b_1 + b_2) = (a_1 + a_2, (a_1 + a_2)s + e_1 + e_2)$. This isn't *quite* a Ring-LWE sample, since the error $e_1 + e_2$ may conspire to be slightly larger than "small". So this method doesn't *quite* work as stated, but if we let the oracle accept slightly larger errors, it would work.

Therefore, on a random challenge, there's a 1/16 probability of error in your response. In other words, you have probability 15/16 of being correct.

Comments: You could improve this by also considering the difference (not just sum) of the samples. There are two small holes in the solution: the issue of three-way correlation, and the issue of the error inflation ($e_1 + e_2$). These are consequences of the way I stated the problem more than being essential.