

HW 8 CMSC 456. DUE Nov 12
NOTE- THE HW IS FIVE PAGES LONG

1. (0 points) READ the syllabus- Content and Policy. What is your name?
Write it clearly. What is the day of the final?

GOTO NEXT PAGE

2. (Read the slides on LWE Diffie-Hellman) (WARNING: this problem continues onto the next page. Write the following programs. I highly suggest using python's numpy library to implement this easier.)

(a) *GENMATRIX*(n, p): gen a rand $n \times n$ matrix of elements of $\{0, \dots, p - 1\}$. We view entries as elements of \mathbb{Z}_p .
(See `numpy.randint` - it can generate random integer arrays. This can be done with one line)

(b) *GENER*(n, p): gen a rand n -vector of elements of $\{0, 1, p - 1\}$.

- Prob of a 0 is $\frac{n-2}{n}$
- Prob of a 1 is $\frac{1}{n}$
- Prob of a $p - 1$ is $\frac{1}{n}$

We view entries as elements of \mathbb{Z}_p .

(You can generate a random floating-point number in the range $[0,1)$ with `numpy.random`)

(c) *GENDATA*(n, p): (This is pseudocode, NOT actual python code — you will have to translate this into actual code)

i. $A := \text{GENMATRIX}(n, p)$

ii. $\vec{y} := \text{GENER}(1/n)$

iii. $\vec{e}_y := \text{GENER}(1/n)$

iv. $\vec{x} := \text{GENER}(1/n)$

v. $\vec{e}_x := \text{GENER}(1/n)$

vi. $a = \vec{y}A\vec{x} + (\vec{y} \cdot \vec{e}_x)$

(`numpy.mod(numpy.dot(\vec{y} , A), p)` can be used to perform a dot product over modulo p)

vii. $b = \vec{y}A\vec{x} + (\vec{x} \cdot \vec{e}_y)$

viii. if $a \in \{0, \dots, \lfloor p/4 \rfloor\} \cup \{\lfloor 3p/4 \rfloor, \dots, p - 1\}$ $\hat{a} = 0$, else $\hat{a} = 1$.

ix. if $b \in \{0, \dots, \lfloor p/4 \rfloor\} \cup \{\lfloor 3p/4 \rfloor, \dots, p - 1\}$, $\hat{b} = 0$, else $\hat{b} = 1$.

x. the variable `agree` is YES if $\hat{a} = \hat{b}$ and NO otherwise.

xi. Your code will output a tuple or an array of $[a, b, \hat{a}, \hat{b}, \text{agree}]$

GOTO NEXT PAGE

xii. Here is a sample of printing your output:

OUTPUT STARTS HERE

$n = 5, p = 17, N = 5.$

a	b	\hat{a}	\hat{b}	agree
3	2	0	0	<i>YES</i>
10	12	1	0	<i>NO</i>
7	9	1	1	<i>YES</i>
1	0	0	0	<i>YES</i>
5	6	0	0	<i>YES</i>

\hat{a} and \hat{b} agree 80% of the time.

END OF OUTPUT

*****Note that $N = 5$ and there are five lines.*****

GOTO NEXT PAGE

NOTE- For the above problems no points are given but submit anyway to help us grade the problems below which ARE for points.

NOTE- THIS IS STILL PROBLEM TWO:

- (a) (0 points- But do it to check your program. Do not give us the output). Run program *GENDATA* with the following inputs.
- i. $n = 4, p = 19$
 - ii. $n = 10, p = 23$
- (b) Make a method to take $[n, p, N]$ as input and output (1) the percent of agreement (called peragree) (2) the percent of the time they agree and the bit is 0 (called peragree0) (3) the percent of the time they agree and the bit is 1 (called peragree1). Call this method *GENDATA2*(n, p, N).
- (c) Make a method to take a LIST of $[n, p, N]$ inputs and output a table of the n, p, N and peragree, peragree0, peragree1. Call this method *GENDATA3*(n, p, N). A sample output is:

n	p	N	peragree	peragree0	peragree1
5	17	5	80	45	55
6	19	5	75	48	52
7	23	10	90	49	51

This can be generated by printing

“ $n \backslash tp \backslash tN \backslash tperagree \backslash tperagree0 \backslash tperagree1$ ”

If you follow this format for the entries of the table, your results should line up.

- (d) (25 points) Run *GENDATA3* using all $5 \leq n \leq 100$ where $n \equiv 0 \pmod{5}$, primes $p \in \{7, 11, 31, 101\}$ and $N = 1000$.
- (e) (5 points) Note the highest and lowest peragree values, as well as the highest and lowest peragree0 values.
- (f) (10 points) Try changing *GENERR* to use the following probabilities instead:
- i. Try (Prob of 0 is $1 - \frac{2}{n^2}$, Prob of 1 and Prob of $p - 1$ are $\frac{1}{n^2}$)
 - ii. Try (Prob of 0 is $\frac{1}{2}$, Prob of 1 and Prob of $p - 1$ are $\frac{1}{4}$)
 - iii. Try (Prob of 0 is $\frac{n-4}{n}$, Prob of 1 and Prob of $p - 1$ are $\frac{2}{n}$)

Rerun *GENDATA3* with the probabilities above. Look at the *peragree* and *peragree0* values for these as well as the original distribution.

Based on this, which distribution do you think works best (has high agreement and *peragree0* close to $\frac{1}{2}$)? Give a brief justification why it's better than the other distributions.

GOTO NEXT PAGE

3. (30 points). We assume the secret is of length n . For the problems below explain it so that someone who has never seen secret sharing can understand it (This is not hypothetical. Two of the TAs do not know secret sharing (except what the goal is). Lets call them J1 and J2. J1 is grading this problem and will learn this protocol from you!)
- (a) (15 points) Describe the random-string $(2, 5)$ secret sharing scheme. You must describe both what Zelda gives out, and how any two people can determine the secret. How many strings does each person get?
 - (b) (15 points) Describe the polynomial $(2, 5)$ secret sharing scheme. You must describe both what Zelda gives out, and how any two people can determine the secret. How many strings does each person get?

4. (30 points) (This is not something I did in class so it may require some more thought.) For the problems below explain it so that someone who has never seen secret sharing can understand it (This is not hypothetical. Two of the TAs do not know secret sharing (except what the goal is). Lets call them J1 and J2. J2 is grading this problem and will learn this protocol from you!)

Zelda has a secret $s \in \{0,1\}^n$. She wants to share a secret with Alice, Bob, Carol, Donna, Edgar, Frank (A,B,C,D,E,F) such that the following happens:

If Alice, Bob and ANY TWO of $\{C, D, E, F\}$ get together then they can find the secret (and of course any superset of that). No other set can find the secret.

Give a scheme that achieves this. The security must be information theoretic. Both say what Zelda does and what the various combinations of people do. Discuss what happens if any set other than those above gets together.