ECE 3170 Cryptographic Hardware for Embedded Systems

Fall 2025

Assoc. Prof. Vincent John Mooney III
Georgia Institute of Technology

Homework 1, 85 pts.

Due Monday August 25 prior to 11:55pm (please turn in homework electronically on Canvas)

For this homework, please list those with whom you have had substantial conversations about the homework, both from the class as well as those not enrolled. Please also list all other sources of information, e.g., books and internet articles. For students enrolled in the course, please give their full names. For anyone you substantially consult about this homework who is not enrolled, please provide the person's first name and initial of the person's last name. This homework will focus on brute force attacks on secret keys.

To anticipate questions regarding "what is substantial help," first of all commenting on the fact that you are working on the homework without discussing technical content is not "substantial." For example, if you tell your parents or friends that you are "working on a homework for a class on cryptographic hardware," clearly such comments are insubstantial regardless of how often or with how many people the comments occur. Please also consider a single question about something mathematical or technological, e.g., asking a friend about some particular mathematics for this homework one time, to not be "substantial" since it is only one question (even though the content is relevant). However, clearly a continued conversation over time (e.g., during a 10-minute meeting or over the course of several days) about technical content would be substantial help.

1) (5 pts.) List those people with whom you have had substantial conversations about this homework; if you have had none, please so state. Furthermore, other than the course textbook and lecture notes, list all sources from which you received substantial help. (An empty answer to this question, question 1, will receive zero points.)

An empty answer received zero points

2) (25 pts.) Consider a brute force attempt to crack keys for the following scenario. You have in your possession a wireless emulator the size of a 4" x 4" x 4" cube (i.e., of size 64 square inches) with a weight of less than 4 pounds which you can take to a parking lot containing a very large number (many thousands and thousands) of new cars. Each car has a three-digit wireless door key (between "000" and "999" using wireless symbols) as well as a four-digit engine key (between "0000" and "9999" including the endpoints, i.e., "0000" is a legitimate engine key as is "9999").
    a. (10 pts.) The first step is to open the door of one of the cars. In order to not lock out a customer due to others using wireless keys nearby, there is

no limit on the number of attempts. Assume that each attempt takes 100 milliseconds and is broadcast to the nearest 10 cars which have not yet been opened. Furthermore, assume that the parking lot is not monitored between 2am and 3:40am, the attack occurs during these 100 minutes, and that the attacker is walking with a new (distinct) set of 10 cars tested by each attempt. How many cars can be broken into by the wireless emulator during these 100 minutes, ignoring the time it takes to walk around the parking lot and ignoring any "reset" time (in other words, a wireless break-in attempt is made every 100 milliseconds without delay between attempts)?

Please note that the following information was given in the problem:

- Probability of successfully opening the door of one car given one key guess

$$\frac{1}{1000}$$

- Each attempt targets 10 cars

From this we can calculate

- Total number of cars attempted

$$100\ minutes * 60\frac{second}{minute} * 10\frac{attempt}{second} * 10\frac{cars}{attempt} = 600{,}000\ cars$$

- Expected number of cars broken into

$$\frac{1}{1{,}000} * 600{,}000 = 600\ cars\ opened$$

Please note that assuming every car was broken into on the first try was not considered a reasonable approach as it trivializes the question.

b. (10 pts.) The second step is to emulate the wireless engine key. However, this involves pressing one's foot on the brakes and pressing a button which takes 5 seconds per attempt. Furthermore, after 1001 failed attempts the engine locks itself, refuses to accept further attempts and triggers an alarm. Therefore, the thief with the wireless emulator makes at most 1000 attempts per car engine. Assuming all cars on the lot (e.g., if there are 3,000 then all 3,000 are unlocked) have been broken into (i.e., their doors are open providing access to the brake and to the ignition button), how many car engines can be started in 100 minutes of malicious effort?

Please note that the following information was given in the problem:

- 100 minutes of malicious attempts

- 5 seconds per attempt
- At most 1000 attempts per engine

Can calculate maximum number of attempts to be

$$100\ minute * 60\frac{second}{minute} * \frac{1\ attempt}{5\ second} = 1200$$

Method #1
Probability of finding the right engine key with one guess is

$$\frac{1\ guess}{10{,}000\ possible\ keys} = 0.0001$$

On average the number of car engines started is

$$1200\ attempts * 0.0001\ chance\ to\ guess\ correctly = 0.12\ cars$$

Method #2
There are 1200 cars available for attack
There are 10000 key combinations to try

Instead of trying at most 1000 key combinations per car, thief may choose 1 key combination on 1200 cars.
This means the expected number of started car engines is

$$1200\ cars * 0.0001 = 0.12\ cars$$

Please note that assuming every engine was broken into on the first try was not considered a reasonable approach as it trivializes the question.

c. (5 pts.) Did you need to make any assumptions in answering either of the above two parts of this problem, problem 2 on homework 1 of ECE 3170? If you did not have to make any assumptions, please explain one way you could have changed the problem, e.g., by adding a limit on failed attempts, and explain how this change would have possibly affected (or not affected) car customers. If you did have to make one or more assumptions, please explain one of the assumptions you made (i.e., in your answer to 2.a or 2.b) as well as an alternative to the assumption (i.e., explain what could have been a different resolution to the lack of clarity which presumably would have changed your answer). Finally, please do not ask if assumptions are needed or not (more specifically, do not ask Professor Mooney or the Graduate Teaching Assistant); deciding / figuring out whether and which assumptions are or are not needed is part of the

homework, and so the answer(s) will only be provided after the homeworks are turned in (please note that there may be multiple correct answers as well as multiple incorrect answers).

For both parts a) and b), one assumption is that the car and engine keys have a uniform distribution. Another assumption in part a) was that whenever a car door is opened, a new car will magically appear to replace the successfully broken into car's place. Note that not all students have stated the above assumptions, but points were given if their assumptions were reasonable.

3) (15 pts.) Now consider a computer program that is encrypted with a 56-bit key. Assume that the decryption algorithm is known. Further assume that the file containing the encrypted computer program can be decrypted in ten thousand clock cycles on a Gigahertz microprocessor (i.e., one clock cycle takes one nanosecond). Further assume that an incorrect decryption is clearly evident at no cost due to the syntax of the result; similarly assume that a correct decryption is clearly evident for the same reason (e.g., the computer program follows C++ syntax rules nearly perfectly). In other words, only account for the different key values and do not worry about the time needed to check the result.
    a. (5) How long does it take to test half of the key values using one computer? Please give your answer in time units of years. Please further assume that all years have 365 days (do not include leap years in your calculation).

- There are 2^{56} possible key values
- The time per key is

$$10^{-9}\frac{second}{cycle} * 10^{4}\frac{cycle}{key} = 10^{-5}\frac{second}{key}$$

- The time needed to test half the key values is

$$\frac{2^{56}}{2} key * \frac{10^{-5}\ second}{key} * \frac{1\ hour}{3600\ second} * \frac{1\ day}{24\ hour} * \frac{1\ year}{365\ day}$$

$$= 11424.65659\ years$$

    b. (5) Assuming the worst case that you have to test all possible key values, and further assuming that you have 5000 computers available for your exclusive use in a large server farm, can you guarantee to decrypt the file in a year or less? Explain your answer please (an answer of "yes" or "no" without any explanation will receive zero credit).

The time needed to test all possible key values is simply the result calculated in a) multiplied by (2/5000), which does not result in less than a year.

No, decryption cannot be guaranteed in a year or less.

c. (5) Regardless of your answer above, how long would it take to test all possible key values given a server farm with 5000 computers? Please give your answer in units of both (i) years and (ii) hours. Please give your answers (i) and (ii) with at least five significant digits each (you may round off your answer).

$$\frac{2^{56}\ key}{5000\ computer} * \frac{10^{-5}\ second}{key} * \frac{1\ hour}{3600\ second} * \frac{1\ day}{24\ hours} * \frac{1\ year}{365\ day}$$

$$= 4.5698\ years$$

$$\frac{2^{56}\ key}{5000\ computer} * \frac{10^{-5}\ second}{key} * \frac{1\ hour}{3600\ second} = 40{,}032\ hours$$

4) (30 pts.) Birthday Attack Suppose we want to calculate the probably that there is at least one occurrence of two (or more!) students in this class (ECE 3170 CHES) having the same birthday. As of today (August 15, 2025), there are approximately 35 students in ECE 3170. What is the probability that there is at least one birthday in common among the students in the class?

a. (5) Describe how you would approach this problem by calculating the probability that no one in the class has the same birthday.

$P(at\ least\ 2\ people\ share\ a\ birthday)$ and $P(no\ one\ shares\ a\ birthday)$ are mutually exclusive events, hence

$$P(at\ least\ 2\ share\ a\ birthday) = 1 - P(no\ one\ shares\ a\ birthday).$$

We can calculate $P(no\ one\ shares\ a\ birthday)$ as the product of each individual student's chance of not having the same birthday as students already in the room. By considering this sequence of students entering one at a time, we arrive at the probability that no one shares a birthday with the previous students, and thus all have unique birthdays.

b. (15) Write a mathematical formula for your calculation. Do not give the final answer or simplify in any significant way (you may use "…" as long as it is clear what would be placed in the "…" location). Professor Mooney will give some hints to help you, so please make sure you have listened to all class lectures provided so far.

$$P\ (at\ least\ 2\ share\ a\ birthday) = 1 - P(no\ one\ shares\ a\ birthday)$$

Following the sequence described in part a,

- The first student (one student total) has a probability of 365 out of 365 (1) that no one else has the same birthday.
- The second student has a chance of 364 out of 365 of not sharing the same birthday as the first student.
- The third student has a chance of 363 out of 365 of not sharing the same birthday as the previous two students.

$$P(no\ one\ shares\ a\ birthday) = 1 * \frac{364}{365} * \frac{363}{365} \ldots \frac{331}{365}$$
$$= \frac{364!}{365^{34} * 330!}$$

$$P\ (at\ least\ 2\ share\ a\ birthday)\ =\ 1\ -\ \frac{364!}{365^{34} * 330!}$$

c. (10) Now give your final answer: what is the probability that there is at least one case with two students or more sharing the same birthday?

81.44%

5) (10 pts.) In the Media Gallery on Canvas, listen to the lectures "Lecture4DES" and "Lecture5RSA." For the lecture on DES, what mistake does Professor Mooney correct in the recording? (HINT: the correction has to do with the key and its permutation.)

Professor Mooney originally had 64, 56, 48, 40, 32, 24, 16, 8, and 0 as the bits missing. But he went on to remove the 0th bit since he said there was no bit 0, because the permutations are labeled from 1 to 64 rather than 0 to 63. This means that the bits that are missing are actually 64, 56, 48, 40, 32, 24, 16, and 8.