Massachusetts Institute of Technology — Handout 2
6.857: Applied Cryptography and Security — Feburary 7, 2021
Professors Ronald L. Rivest and Yael Tauman Kalai — **Due:** February 22, 2021

Problem Set 1

This problem set is due on *Tuesday, February 22, 2022* at **11:59 PM**. Please note our late submission penalty policy in the course information handout. Please submit your problem set, in PDF format, on Gradescope. *Each problem should be in a separate PDF.* Have **one and only one group member** submit the finished problem writeups. Please title each PDF with the Kerberos of your group members as well as the problem set number and problem number (i.e. *kerberos1_kerberos2_kerberos3_pset1_problem1.pdf*).

You are to work on this problem set in groups. For problem sets 1, 2, and 3, we will randomly assign the groups for the problem set. After problem set 3, you are to work on the following problem sets with groups of your choosing of size three or four. If you need help finding a group, try posting on Piazza or email `6.857-staff@mit.edu`. You don't have to tell us your group members, just make sure you indicate them on Gradescope. Be sure that all group members can explain the solutions. See Handout 1 (*Course Information*) for our policy on collaboration.

Homework must be submitted electronically! Each problem answer must be provided as a separate pdf. Mark the top of each page with your group member names, the course number (6.857), the problem set number and question, and the date. We have provided templates for LaTeX and Microsoft Word on the course website (see the *Resources* page).

Grading: All problems are worth 10 points.

With the authors' permission, we may distribute our favorite solution to each problem as the "official" solution—this is your chance to become famous! If you do not wish for your homework to be used as an official solution, or if you wish that it only be used anonymously, please note this in your profile on your homework submission.

Problem 1-1. Security Policy for AirTags

An AirTag is a small device that helps users find their lost personal objects. The AirTag emits a Bluetooth signal that gets picked up by nearby devices, which help locate the tag using Apple's Find My protocol (`https://en.wikipedia.org/wiki/Find_My`). There is no need to understand this protocol for the sake of this problem.

In this problem, we will explore some of the security principles we discussed in class by designing a **security policy** for Apple's AirTags. Recall that a security policy is a description of what parties are (not) supposed to do, usually written in the form of *principals* performing *(im)permissible actions* on *objects.* For example, in a computer system, an entry in its security policy could be "only the creator of a file is allowed to change its permissions".

To help you better understand how AirTags work, you can take a look at this article: `https://www.fastcompany.com/90628073/apple-airtag-privacy-security`

Also, for simplicity, you can assume that AirTags can get paired up with any phone (i.e., there is no distinction between, say, iOS and Android phones). You can add other constraints/assumptions if you feel the need, but please state these very clearly!

(a) Define the parties involved in the system.

(b) Write the desired functionalities of an AirTag (i.e., what "good" things it should do, if no adversaries are involved).

(c) Write a security policy for the AirTags. Make sure to refer back to the principals you defined in your first answer. In addition, explain how your security policy relates to the **security goals** we discussed in the first lecture (the "CIA" triad – Confidentiality, Integrity, Availability). That is, for every statement in your security policy, state the security goal under which it falls.

(d) Mention three of the AirTags' **security mechanisms**, and explain how they help enforce your security policy. For each mechanism, state if it is a "prevention" or "detection" mechanism.

(e) Note that protecting against theft of personal items is **not** a security goal of the AirTags. Why do you think this is the case? Hint: theft protection is in opposition with one of the AirTag's privacy goals (see the linked article for more help).

This is a very open-ended problem, and there is no "right" solution.

Problem 1-2. Breaking a "multi-time pad" cipher

Alice and Bob are part of a book club. They've started reading this week's new book ahead of everyone else and they can't wait to discuss it! Alice wants to send Bob an excerpt she thinks will be important later in the book, but they don't want to spoil it for the other members of the club...

Since both of them are taking 6.857, they decide to come up with a secure scheme to encrypt messages so that Alice can send the excerpt to Bob securely. The scheme is described below.

Some definitions:

Key: a key is a fixed-size sequence $K = (k_1, k_2, \cdots, k_{32})$ of 32 5-bit unsigned integers.

Encryption table: an encryption table F is a fixed public 32×32 table where each each integer v, such that $0 \leq v \leq 31$, appears exactly once in each row and column. Rows and columns are indexed from 0 to 31 and we denote the value on row x and column y by $F(x, y)$.

The scheme:

1. At the first book club meeting, Alice and Bob decide on a secret key K (which no one else knows about). (They also agree to use the public encryption table F given to them in 6.857.)
2. Later in the week, Alice chooses a book excerpt $T = (t_1, t_2, \cdots, t_n)$ where each t_i is a character. She then encodes each character t_i as a 5-bit unsigned integer m_i. There are only 32 characters allowed: lowercase letters $a - z$, which are mapped to 0-25 in order; spaces, periods, commas, exclamation points, question marks, and new-line characters, which are mapped to 26-31 in this order. (Alice's original text only has allowed characters).
3. She then computes the encrypted text $C = (c_1, c_2, \cdots, c_n)$, where c_i is defined as:

$$c_i = F(k_i, m_i)$$

The key indices are taken modulo 32, i.e., $k_{33} = k_1, k_{34} = k_2$, etc. (not exactly mod 32, which has a range from 0 to 31, but one more: this has a range from 1 to 32. But the effect is essentially the same).

4. Finally, she sends C to Bob.

(a) Show how Bob can obtain the original text T by using C, K, and F.

It's almost time for the club meeting and Eve hasn't started reading the book! She thinks Alice's excerpt will tell her something important about the story, so she decides to break Alice and Bob's code. However, she only has access to their algorithm and encryption table F and knows nothing about their secret key.

(b) Similarly to the one-time-pad we saw in lecture, if the number of characters in Alice's text is less than or equal to 32, then Eve can extract no information about the text T from the encrypted text C. Fortunately for Eve, the text was quite long. Your task is to help her retrieve the original text from the ciphertext C she overheard. We have provided you with:

- **encryption_table.txt**: the public encryption table F. Rows and columns are indexed from 0 to 31 top to bottom and left to right, respectively;

- **ciphertext.txt**: the ciphertext C – a sequence of numbers between 0 and 31 encoding their respective characters in the encrypted text;
- **frequencies.txt**: the list of approximate frequencies of characters. On row i, you should find the approximate frequency that the character mapped to i appears on the book.

What was the original text? (Hint: for each position, try to identify the key value that maximizes the joint probability of the deciphered ciphertext letters.) Describe your strategy and submit your code and solution. How much ciphertext was needed to find the solution?

Problem 1-3. Signal Messenger

Signal messenger is an end-to-end encrypted messaging application. End-to-end encryption means that when two parties send messages over Signal, the contents of those messages are hidden from all intermediate parties and are only visible to the sender and recipient of the messages. Importantly, even Signal's servers used to transfer messages cannot read their contents.

However, the confidentiality provided by Signal's encryption is not the only property that Kyle and Andres want for their messages! They also want *authenticity.* That is, Kyle wants to be confident that she is really talking to Andres and not some imposter.

To achieve this, Signal has implemented *safety numbers.* The safety number is a per-conversation identifier where a *conversation* is the exchange of messages between two signal accounts. The safety number is defined based on the two endpoints of that conversation where $endpoint_1$ is the initiator of the conversation and $endpoint_2$ is the responder.

$$safenum_{(endpoint_1, endpoint_2)} = f(endpoint_1, endpoint_2)$$

In practice, f is a hash function and each *endpoint* is the public key of one of the conversation partners (though you do not need this information to solve this question). What is important to understand, is that safety numbers are *unique* in that no two conversations on the Signal application will ever share the same safety number.

This property allows Signal users like Kyle to *recognize* their conversation partners. A conversation between Kyle and Andres will have a safety number that Kyle can distinguish from that of a conversation with any other Signal user. The safety number for a conversation will change if either endpoint of the conversation changes from what it was at the beginning of the conversation.[1]

It is important to recognize here that *endpoint* refers to the Signal application of a person's phone (as it is receiving the messages, generating the safety number, etc.) not to the person reading the messages or to their account. (Because of this, reinstalling the Signal application is a change in endpoint and will result in a changed safety number for the conversation.) Reinstalling the application will, however, **not** change the person involved or their account and one person/account may thus end up many different instances of their endpoint and, correspondingly, safety numbers[2] for their conversations over time.

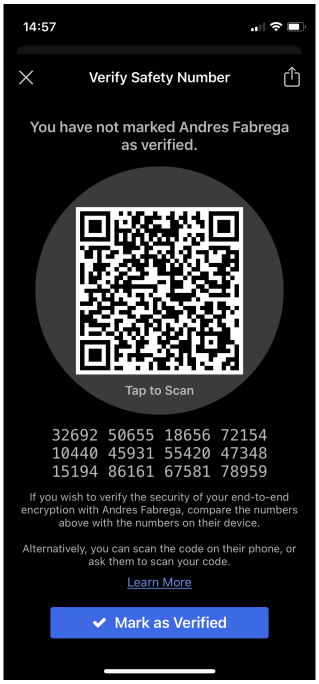
In the event that the safety number for a conversation changes, Signal will notify both users as this indicates that one of the two endpoints has changed from what it was at the beginning of the conversation.

Linking the endpoint to the person who owns it is the point of safety numbers and what we will focus on in this question.

Because the safety number represents the conversation, both participants in the conversation can meet in person to *verify* the safety number for their conversation by simply checking that they each see the same number on their device:

[1] Note, however, that the conversation itself is *persistent* as it is tied to the accounts involved, not their endpoints.

[2] because safety numbers are computed based on the endpoint, not the person/account involved



14:57

You have not marked Andres Fabrega as verified.

```
32692 50655 18656 72154
10440 45931 55420 47348
15194 86161 67581 78959
```

If you wish to verify the security of your end-to-end encryption with Andres Fabrega, compare the numbers above with the numbers on their device.

Alternatively, you can scan the code on their phone, or ask them to scan your code.

For more information about safety numbers you can read this blog post by Signal: `https://signal.org/blog/safety-number-updates/`

For the sake of this question you can assume that Kyle, Andres, and Lucas are all **honest** and will not collude with adversaries like Eve[3].

(a) Kyle wants to use Signal to plan this assignment with Andres. She's pretty sure she remembers his phone numbers so she opens up a new conversation on Signal to message him and gets a reply: "Hi Kyle, this is Andres!". Kyle still isn't positive she got Andres' phone number right and is worried that she might actually be talking to an imposter, Eve, who is trying to learn the problem solutions! She notices that that there has been **no** change in the safety number since the beginning of the conversation, but is not reassured. Why isn't the unchanged safety number enough to convince her that she is talking to Andres?

(b) Instead, Kyle and Andres decide to meet in person to verify that their safety numbers match. When they meet, Kyle finds that the safety number provided by Andres **does** indeed match that of the Signal conversation she has been having. Should she be convinced of the identity of her conversation partner now? Why or why not? Does it matter whether Kyle and Andres are strangers who meet for the first time to perform this verification?

(c) Now consider several scenarios for verifying safety numbers and mark them as *secure* or *insecure*. Explain your reasoning.

1) Kyle and Andres have never met before, but Lucas knows both of them. They all meet in person and Lucas introduces Kyle and Andres to each other before they verify that their safety numbers are equal.

2) While messaging over Signal, Kyle is notified that the safety number for her conversation with Andres has changed. She messages to ask what happened and gets the response that Andres just reinstalled the app including a screenshot of the new safety number so she can verify it.

[3] who is an *EVE*sdropper

3) Kyle and Lucas are starting a new conversation and need to verify their safety number, but it's cold outside and they don't want to meet in person. Since they already do a lot of Zoom calls, they decide to verify remotely by each holding up the QR code of their safety number and reading the number out loud.