Foundations of Computer Security September 16, 2022
Massachusetts Institute of Technology 6.1600 Fall 2022
Henry Corrigan-Gibbs, Yael Kalai, Nickolai Zeldovich Problem set 1

Problem set 1

This assignment is due **at 10:00pm ET** on **Thursday, September 29, 2022**.

Please make note of the following instructions:

- Remember that your solutions must be submitted on Gradescope. Please sign-up for 6.1600 Fall 2022 on Gradescope, with the entry code `577GE7`, using your MIT email.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available on the course website (`https://61600.csail.mit.edu/2022/`). Each submitted solution should start with your name, the course number, the problem number, the date, and the names of any students with whom you collaborated.

Problem 1-1. Hash Function Properties [50 points]

Let $h : \{0,1\}^{\leq 2n} \rightarrow \{0,1\}^n$ be a hash function that is collision resistant. Let $h' : \{0,1\}^{\leq n+1} \rightarrow \{0,1\}^{n+1}$ be the hash function given by the rule

$$h'(x) = \begin{cases} 0||x & \text{if } x \in \{0,1\}^n \\ 1||h(x) & \text{otherwise} \end{cases}$$

(a) Prove that h' is not one-way. [15 points]

Definition 1 *A function $f : X_n \rightarrow Y_n$ is said to be one-way if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$, on input n and $y = f(x)$ for a random $x \in X_n$, outputs any x' such that $f(x') = y$, is negligible.*

(b) Prove that h' is collision resistant. [20 points]

Definition 2 *A function $f : X_n \rightarrow Y_n$ is said to be collision resistant (CR) if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$ on input n, outputs any distinct $x, x' \in X_n$ such that $f(x) = f(x')$, is negligible.*

(c) Prove that h' is target collision resistant if h is target collision resistant. For this problem part, assume that h has the form $h\colon \{0,1\}^{\leq n+1} \rightarrow \{0,1\}^n$. [15 points]

Definition 3 *A function $f : X_n \rightarrow Y_n$ is said to be target collision resistant (TCR) if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$ on input n and a random $x \in X_n$, outputs $x' \in X_n$ such that $x' \neq x$ and $f(x) = f(x')$, is negligible.*

Problem 1-2. Message Authentication [50 points]

The material for this problem will be covered in class on Monday, September 19th.

(a) Let MAC be a secure message authentication code. Suppose Alice and Bob send authenticated messages to each other. Namely, every time one of them sends a message M they send it together with $\mathsf{MAC}(K, M)$ where K is their shared secret key. On day 1, Alice asks Bob if he wants to go to the movies, and Bob replies "yes". On day 2, Alice asks Bob if he wants to go to ice cream and Bob replies "no". On day 3, Alice asks Bob if he wants to rob a bank and Bob replies "no". Can an adversary Eve observing the communication on the first two days, corrupt Bob's message on the third day (in an authenticated way)? If so, how would you use a secure MAC so that the adversary cannot corrupt Bob's message? [15 points]

(b) Recall that the CMAC construction we saw in class is a sequential construction. Namely, to MAC a very long message that consists of L blocks (each of 128 bits), we need to do L sequential steps. Consider the following parallel construction: Let $F : K \times X \to \{0,1\}^k$ be a pseudorandom function (PRF). Let

$$\mathsf{MAC}(K, (M_1, \ldots, M_L)) = \oplus_{i=1}^{L} F(K, M_i),$$

where each $M_i \in X$. Is this a secure MAC (i.e., existentially unforgeable against adaptive chosen message attacks)? [15 points]

(c) Recall that to apply the CMAC construction we need to assume that the message length is a multiple of 128, since we partition the message into blocks, each of length 128. If the length of the message is not a multiple of 128 then we need to pad it to ensure that it's length is divisible by 128.

- Consider the padding which simply pads the message M with 0's to make it of length that is divisible by 128. Argue that the resulting (CMAC+padding) scheme is insecure by providing an attack.
- Suggest a padding scheme that will make the resulting (CMAC+padding) scheme secure. What property does such a padding scheme need to have in order to ensure that the resulting scheme is secure? **Hint:** You may need to add a dummy block. [20 points]

Problem 1-3. Authenticating a photo log [50 points]

In Lab 1, you must design an authentication scheme for an operation log. In particular, a user has many devices, all sharing a 128-bit secret key, that interact via a potentially adversarial server. (See the Lab 1 instructions for details.)

(a) Ideally, we would like that if one device uploads a photo to the log on the server, eventually every other device sees that photo in its operation log. Explain why such a strong security property is unachievable in this setting.

(b) Explain why, if the server has unbounded computational power, after observing a small constant number of authenticated log entries, it can trick an honest device into accepting a log entry that the user did not add.

(c) One way to authenticate the operation log is to:

- concatenate the variable-length byte arrays representing the photos in each log entry into one big string,
- hash the big string using a collision-resistant hash function, and then
- compute a MAC of the hashed value.

The device would then upload the resulting MAC tag to the server.

Show that if the devices use this authentication strategy, the server can trick a honest device into accepting a forged operation log.

(d) Explain one way to fix the problem of Part (c).

(e) Simple techniques for authenticating an operation log of L entries require the device to perform $\approx L$ cryptographic operations (hashes and/or MACs) every time it synchronizes with the server. Modify your solution in Part (d) to require only $O(U)$ cryptographic operations after each synchronization, where U is the number of log entries the server sends that the receiving device has not already seen. (In other words, your update time should scale with the number of *new* log entries, rather than the number of *total* log entries.)

If your solution to Part (c) already has this property, then just explain why it does.

(f) Modify your solution from Part (d) so that the number of cryptographic operations the receiving device performs scales linearly with the number of *new and valid* log entries. That is, a malicious server should not be able to trick the client into performing extra cryptographic operations. If the device ever detects server misbehavior during synchronization, it may return immediately without updating its state.

If your solution to Part (d) already has this property, then just explain why it does.

Problem 1-4. EXTRA CREDIT (CHALLENGING!): Collisions [5 points]

Warning: The extra-credit problems in this course are challenging and are worth very few points. So please only attempt them if you have completed the rest of the problem set and are looking for more.

This problem makes use of a hash function $H\colon \{0,1\}^n \to \{0,1\}^n$, which you should think of as a truly random function (i.e., a random oracle). That is, for every $x \in \{0,1\}^n$, think of the value $H(x)$ as a bit string sampled independently and uniformly at random from $\{0,1\}^n$.

Define the function $\hat{H}\colon \{0,1\}^{4n} \to \{0,1\}^n$ as:

$$\hat{H}(x_1, x_2, x_3, x_4) := H(x_1) \oplus H(x_2) \oplus H(x_3) \oplus H(x_4).$$

Show that it is possible to find four distinct strings x_1, x_2, x_3, x_4 such that $\hat{H}(x_1, x_2, x_3, x_4) = 0^n$ in time $2^{n/3} \cdot \mathsf{poly}(n)$.