Massachusetts Institute of Technology
6.5610: Applied Cryptography and Security
Professors Henry Corrigan-Gibbs and Yael Tauman Kalai

Handout 3
February 27, 2023
Due: March 10, 2023

Problem Set 2

This problem set is due on *Friday, March 10, 2023* at **4:59 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 page.*

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. See the course website for our policy on collaboration. Each group member must independently write up and submit their own solutions.

Homework must be typeset in LaTeX*and submitted electronically!* Each problem answer must be provided as a separate page. Mark the top of each page with your group member names, the course number (6.5610), the problem set number and question, and the date. We have provided a template for LaTeX on the course website (see the *Psets* tab at the top of the page).

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 2-1. ElGamal variations

Consider the following two variants on the ElGamal encryption scheme, which use the same key generation algorithm we saw in class, but different encryption algorithms:

- $\mathsf{Enc}_1(pk, m)$: The message space is the group $\mathbb{G}$, and to encrypt $m \in \mathbb{G}$ sample a random $r \leftarrow \{1, \ldots, |\mathbb{G}|\}$, and output $(g^r, pk^r \cdot m) \in \mathbb{G}^2$ as the ciphertext.
- $\mathsf{Enc}_2(pk, m)$: Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function. The message space is $\{0,1\}^\ell$, and to encrypt $m \in \{0,1\}^\ell$ sample a random $r \leftarrow \{1, \ldots, |\mathbb{G}|\}$, and output $(g^r, H(pk^r) \oplus m)$ as the ciphertext.

Decide whether each of the following is True or False, and explain why.

(a) Enc_1 is CPA secure under the CDH assumption.

(b) Enc_1 is CPA secure under the DDH assumption.

(c) Enc_2 is CPA secure assuming H is a OWF and assuming CDH.

(d) Enc_2 is CPA secure assuming H is modelled as a random oracle and assuming CDH.

(e) Enc_2 is CCA secure assuming H is modelled as a random oracle and assuming CDH.

Problem 2-2. Encrypt-then-MAC

We discussed in class the importance of Encrypt-then-MAC for authenticated encryption. This problem will explore the security of other variants.

Assume for this problem that $\mathsf{MAC}(k, m)$ is a secure MAC scheme against adaptive chosen message attacks, and $\mathsf{Enc}(k, m), \mathsf{Dec}(k, c)$ is a CPA-secure encryption scheme. Each problem part provides a new encryption scheme $(\mathsf{Enc}', \mathsf{Dec}')$. Say whether the new scheme is necessarily an authenticated encryption scheme under the defintion presented in class. If yes, explain why in a few sentences, and if not, provide an attack.

(a) $\mathsf{Enc}'((k_1, k_2), m)$: return $(\mathsf{Enc}(k_1, m), \mathsf{MAC}(k_2, m))$
$\mathsf{Dec}'((k_1, k_2), (c, t))$: $m = \mathsf{Dec}(k_1, c)$; if $t = \mathsf{MAC}(k_2, m)$, return m; else, return $\perp$

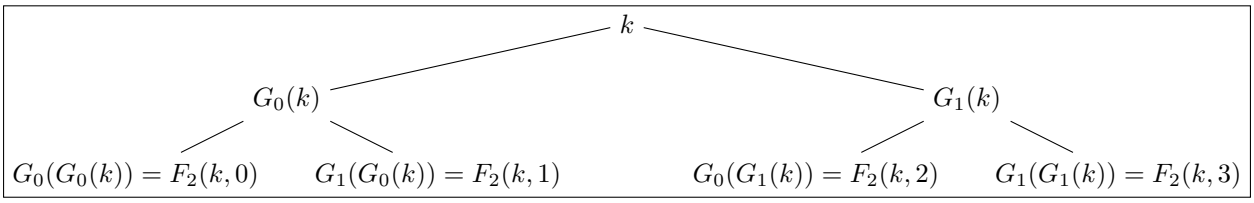

Figure 1: Diagram of the "tree" of evaluations of F_2, on key $k \in \{0,1\}^n$.

(b) $\mathsf{Enc}'((k_1,k_2),m)$: $t = \mathsf{MAC}(k_2,m)$; return $\mathsf{Enc}(k_1, m||t)$
$\mathsf{Dec}'((k_1,k_2),c)$: $m||t = \mathsf{Dec}(k_1,c)$; if $t = \mathsf{MAC}(k_2,m)$, return m; else, return $\perp$

(c) $\mathsf{Enc}'(k,m)$: return $(\mathsf{Enc}(k,m), \mathsf{MAC}(k,c))$
$\mathsf{Dec}'(k,(c,t))$: if $t = \mathsf{MAC}(k,c)$, return $\mathsf{Dec}(k,c)$; else, return $\perp$

(d) $\mathsf{Enc}'(k_1,k_2,m)$: $c = \mathsf{Enc}(k_2,m)$; return $(c, \mathsf{Enc}(k_2, \mathsf{MAC}(k_1,c)))$
$\mathsf{Dec}'(k_1,k_2,(c,t))$: if $\mathsf{Dec}(k_2,t) = \mathsf{MAC}(k_1,c)$, return $\mathsf{Dec}(k_2,c)$; else, return $\perp$

Problem 2-3. Building and constraining PRFs.

In this problem, we will explore how to build a PRF from a PRG. Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a length-doubling PRG. For ease of notation, we write $G(x) = G_0(x)||G_1(x)$, where $G_0 : \{0,1\}^n \to \{0,1\}^n$ and $G_1 : \{0,1\}^n \to \{0,1\}^n$. We also write $i_1, i_2, \ldots, i_m$ to be the m bits in the binary representation of any $i \in \{0,1\}^m$. Then, we define the function $F_m : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$, which consists of m sequential evaluations of either G_0 or G_1:

$$F_m(k,i) = G_{i_m}(\cdots(G_{i_2}(G_{i_1}(k))\cdots)).$$

In other words, we can think of evaluating F_m by assigning "labels" to the nodes of a binary tree of depth m. The root node of this tree is labelled with the value k. For any node with label x, we label its left child with value $G_0(x)$ and its right child with value $G_1(x)$. Then, after propagating labels from top to bottom through the tree, we get that the i-th leaf node is labelled with value $F_m(k,i)$. An illustration of this tree for $m = 2$ is given in Figure 1.

(a) What is the time to evaluate $F_m(k,\cdot)$ at a single point $i \in \{0,1\}^m$, in terms of m and the time to evaluate G?

(b) Suppose that F_m is a PRF, for some $m \geq 1$. Explain in 2-3 sentences whether F_{m+1} a PRF.

(c) Alice wants to communicate the function $F_m(k,\cdot)$ to Bob. Suppose Bob knows m. What is the smallest number of bits Alice can send Bob, so that Bob can recover $F_m(k,\cdot)$ (i.e., Bob can evaluate $F_m(k,\cdot)$ at all points in $\{0,1\}^m$)?

(d) Suppose now that Alice again wants to communicate $F_m(k,\cdot)$ to Bob, *except* she wants to prevent Bob from evaluating $F_m(k,\cdot)$ at a fixed point $i \in \{0,1\}^m$. Suppose Bob knows m and i. What is the smallest number of bits Alice can send Bob to achieve this?
Hint: It will be useful to think of the tree structure of F_m.

Problem 2-4. Programming: Birthday attack

Note: The starter code for this problem is posted on Piazza.

(a) Collisions, birthday paradox. Assume that each student at MIT gets a uniformly random 9 digit student ID number. Find the ith line of `inputs.txt`, where i is your MIT ID number modulo B where $B = 1000$. Assume there are 83 students in this class. What is the probability at least two students have the same line of the table (the same i)? Express your answer as a decimal with 4

significant figures; any code for this part does not need to be attached. If you were given $r > B$ uniformly random IDs, and a table with B entries, what data structure could you use to efficiently find such a collision in (on average) $O(\sqrt{B})$ time and space?

(b) Short discrete log. The number x found on the ith line of `inputs.txt` (where i is your MIT ID modulo 1000) has a discrete log with respect to the generator g in the group $\mathbb{Z}_p^*$. The exact values of g and p can be found in the skeleton code file `dl.py`. Your task is to find the discrete log m such that $g^m = x \pmod{p}$. This task is quite hard when x is large. However, Bob simply computed g^m for a *small* 48-bit message and thought that m was completely hidden. Therefore, you know that the discrete log of x is only 48 bits (e.g $0 \leq m < 2^{48}$). Your attack should somehow exploit the fact that the discrete log x is small. Use the collision/birthday paradox algorithm from part a to find the message (discrete log) m for your input x. Provide m, in base 10, as the answer to this question and attach your code on gradescope.