Prof. Christian Cachin
Luca Zanolini, Ignacio Amores Sesar

Exercise 4

4.1 Deterministic libraries (3pt)

A *deterministic* program is one that uses no random choices. Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are two deterministic libraries with a common interface. Show that either $\mathcal{L}_1 \equiv \mathcal{L}_2$, or else $\mathcal{L}_1$ & $\mathcal{L}_2$ can be distinguished with advantage 1.

4.2 Hash-function collisions (2pt)

Consider an ideal hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$, which can modeled as follows:

$\mathcal{L}_{\text{idealhash}}$
T := [ ]
$\text{QUERY}(x \in \{0,1\}^*)$:
if $T[x]$ is undefined:
$\quad T[x] \leftarrow \{0,1\}^\lambda$
return $T[x]$

a) For output length $\lambda = 40$, estimate the probability of finding a collision if one computes 10^6 hashes on arbitrary, distinct inputs.

b) For $\lambda = 256$ (SHA-256), estimate the number of hashes we need to compute to find a collision with probability $\frac{1}{2}$.

4.3 Salt (2pt)

Many computer systems use passwords for user authentication. It is common practice that the hash of a password is stored on a server instead of the clear-text password. This prevents that passwords are exposed directly in case the server is compromised.

The number of words in a language like German or English is estimated to be a few hundred thousand, but the active vocabulary is smaller than $100'000$ words. Many users select a password from the vocabulary of their language and typically add a digit or two. To be concrete, assume below that every user selects a password randomly among 2^{20} (roughly $1'000'000$) words.

a) Give an estimate on the number of users that are needed such that the password database contains two equal hashed passwords with probability $\frac{1}{2}$ or more. Assume that the hash function is perfect, that is with no collisions.

b) *Salt* is a random data that enters the hash calculation of the password, but which is stored together with the password hash by the server. Hence, a typical password file contains an entry of the form

$$(\textit{username}, \textit{salt}, h)$$

for each user, where $h = H(salt \| password)$ represents a hashed and "salted" password. Assume *salt* is a random 256-bit value. What is now the estimated minimum number of users such that the password database contains two entries that have the same h values with probability $\frac{1}{2}$ or more? Assume here that the hash function is perfect, i.e., that no collisions occur.