# Homework 1

Michele Laurenti - 1603064

January 3, 2017

## 1 Perfect Secrecy and One-Time Pad

a. The encryption scheme would not be perfectly secure, since $\Pr[M = m | C = m] = 0$ for any $m$, in contradiction with the definition of perfect secrecy, which states that $\Pr[M = m | C = c] = \Pr[M = m]$ (we'd have that all messages have probability 0, which does not make sense in a discrete setting - i.e. $\mathcal{M}$ finite).

b. Perfect secrecy means that for any message $m$ $\Pr[M = m] = \Pr[M = m | C = c]$. Consider a distribution $M$ for which exist two distinct messages $m, m'$ such that $\Pr[M = m] \neq \Pr[M = m']$. This implies, using the definition of perfect secrecy, that $\Pr[M = m | C = c] = \Pr[M = m] \neq \Pr[M = m'] = \Pr[M = m' | C = c]$, which contradicts the statement.

c. Since the adversary is given $m$ and $\phi$ in the MAC setting, the adversary could access $k = m \oplus \phi$, and thus compute, for any $m'$, $\phi' = m \oplus k$, which is always authenticated with key $k$. This is not statistically secure, since the adversary produces a valid message and validator pair with probability 1 (1-statistical security is just no security).

## 2 Universal Hashing

a. A family $\mathcal{H}$ is pairwise independent if for all two distinct messages $m, m'$ the couple $(h_k(m), h_k(m')) \in \Phi^2$ is uniform over the (random) choice of $k$. The probability of a collision, given $m, m'$ and a random $s$, can be computed exactly in the case of a pairwise independent function, as

$$\Pr[h_s(m) = h_s(m')] = \sum_{\phi \in \Phi} \Pr[(h_s(m), h_s(m')) = (\phi, \phi)] = |\Phi| \cdot \frac{1}{|\Phi|^2} = \frac{1}{|\Phi|}$$

which gives us the definition of universal hash function, satisfied with an equality.

b. We can think of the proposed hash function as a set of linear applications $h_s : \mathbb{F}^t \to \mathbb{F}$ over vector field $\mathbb{F}^t$. If two "vectors" (i.e. messages) $x$ and $\hat{x}$ collide, then it must be that

$$h_s(x) = h_s(\hat{x}) \implies h_s(x - \hat{x}) = \underline{0}_t$$

From linear algebra we know that the dimension of the kernel of $h_s$ is $t - 1$. The number of vectors inside the kernel of $h_s$ can be upper bounded by

$$|\ker h_s| \leq |\mathbb{F}|^{t-1}$$

Finding a collision $\hat{x}$ with a vector $x$ is the same as finding a vector $z$ inside the kernel of $h_s$, since $\hat{x} = x - z$. Thus the probability of a collision is

$$\Pr[h_s(x) = h_s(\hat{x})] = \Pr[z \in \ker h_s] = \frac{|\mathbb{F}|^{t-1}}{|\mathbb{F}^t|} = \frac{1}{|\mathbb{F}|}$$

Since the co-domain of $h_s$ is $\mathbb{F}$, the proposed hash is universal.

# 3 One-Way Functions

a. An inefficient attacker, given $y = f(x)$ for some $x$, can always try out each possible string in $\{0,1\}^n$ until it finds $x'$ such that $f(x') = y$. Instead, an efficient attacker could sample $\{0,1\}^n$ uniformly at random, obtaining a string $x'$. The probability of success for this attacker can be bounded as

$$\Pr\left[x' \in f^{-1}(y)\right] \leq \Pr\left[x' \in \{x\}\right] = \frac{1}{2^n}$$

where $f^{-1}(y)$ is the set of pre-images of $y$ with respect to $f(\cdot)$, which contains at least $x$.

b. (i) This is not a one way function: given $z = f(x,y)$ the adversary could compute $z_1||z_2$, with $|z_1| = |z_2| = n$ (assuming $f : \{0,1\}^{2n} \to \{0,1\}^{2n}$), by taking $z_1 = \left\lfloor \frac{z}{2} \right\rfloor$ and $z_2 = \left\lceil \frac{z}{2} \right\rceil$, interpreting $z, z_1, z_2$ as binary encodings of natural numbers.

(ii) This is a one way function: assume $\mathcal{A}$ breaks $g$, we can then build $\mathcal{A}'$ which breaks $f$. $\mathcal{A}'$ receives $y = f(x_1)$ for some $x_1$, creates a $x_2$ and gives the message $y||x_2$ to $\mathcal{A}$. Then $\mathcal{A}$ outputs $x' = x'_1||x'_2$ (with $x'_2 = x_2$) with non-negligible probability. Since $g(x') = f(x'_1)||x'_2$, $\mathcal{A}'$ can break $f$ using $x'_1$.

(iii) This is a one way function. With the same argument as given above, given an adversary capable of breaking $g(\cdot)$ we are able to break $f(\cdot)$.

(iv) This is a one way function. Assume $\mathcal{A}$ breaks $g$, we could then build $\mathcal{A}'$ which breaks $f$ with non-negligible probability. $\mathcal{A}'$ receives $y = f(x)$ for some $x$. If $x = \hat{x}||0$, $\mathcal{A}$ will be able to break $g(\hat{x}) = y$ and return $x'$ such that $f(x'||0) = y$. Then $\mathcal{A}'$ would be able to break $f(\cdot)$ using $x'||0$. If $x = \hat{x}||1$, $\mathcal{A}$ could be unable to break it, but that happens only half of the times if $x$ is sampled uniformly. Notice that if $\exists x_1 = \bar{x}||0$ such that $f(x_1) = y$, $\mathcal{A}$ can still be able to break it. Thus $\mathcal{A}'$ can break $f(\cdot)$ with probability at least $\frac{1}{2}$.

# 4 Pseudorandom Generators

a. The exponential time distinguisher could ask for $2^{2\lambda}$ strings (of length $2\lambda$) to the unknown generator. The set of strings that a PRG outputs has size at most $2^\lambda$, i.e. if the distinguisher is querying a PRG it wouldn't get more than $2^\lambda$ different strings. This happens only with a very low probability when querying a uniform distribution of strings of length $2\lambda$. Thus the exponential-time distinguisher could ask for exponentially many strings: if the set of obtained strings has more than $2^\lambda$ strings, then with probability 1 the distinguisher is dealing with a uniform distribution of strings of length $2\lambda$; otherwise, with probability almost 1 it's dealing with a PRG.

b. (i) This is a PRG. Without loss of generality, consider $G'(s_1||s_2) = G(s_1)||G(s_2)$ (the argument can be extended to the general case) with $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+l}$ being a PRG (thus $G' : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda+2l}$). If an adversary could distinguish $G'$ from a distribution $U_{2\lambda+2l}$, we could create an adversary distinguishing $G$ from a distribution $U_{\lambda+l}$. This adversary would take two strings from the unknown distribution, contatenate them, and give them to the distinguisher of $G'$, repeating as needed, and then output the same as the distinguisher.

(ii) This is not a PRG. As the hints says, the proposed construction does not work for all PRG. The function $G(s_1||s_2) = \hat{G}(s_2)$ is a PRG if $\hat{G}$ is a PRG (with $|s_1| = |s_2| = \lambda$). This follows from the fact that a uniform distribution of strings of length $\lambda$ is indistinguishable from the uniform distribution of strings of length $2\lambda$ after discarding the first half of the string. Thus, if $f$ is the function that discards the first half of its input,

$$\hat{G}(U_\lambda) \approx_c U_\lambda \approx_c f(U_{2\lambda}) \approx_c \hat{G}(f(U_{2\lambda})) \approx_c G(U_{2\lambda})$$

The proposed construction of $G'$ is a constant function when built from $G$, thus it's not a PRG.

# 5 Pseudorandom Functions

a. This is not a pseudo-random function. An adversary could query the unknown function $f$ for the values $f(x_1), f(x_2)$ of two distinct $x_1, x_2$, and then check wether $f(x_1) \oplus f(x_2) = x_1 \oplus x_2$. If it's not the case, then the function is a random function with probability 1. If that is the case, with probability almost 1 the function is the PRF.

b. This is not always a PRF. Consider a PRF family $\mathcal{F}$, and the PRF family $\mathcal{F}'$ obtained from $\mathcal{F}$ by chosing a key $k_1$ and by replacing $F_{k_1}$ with the identity function (i.e. $F_{k_1}(x) = x$ in $\mathcal{F}'$). $\mathcal{F}'$ is still a PRF, since if keys are chosen uniformly, the key $k_1$ is chosen with probability $2^{-\lambda}$ (with key space $\{0,1\}^\lambda$). The probability of an adversary of distinguishing $\mathcal{F}'$ from a true random function is

$$\begin{aligned}
&|\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1] - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]| = \\
&|\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 | k = k_1] + \Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 | k \neq k_1] - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]| \leq \\
&|\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 | k = k_1]| + |\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 | k \neq k_1] - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]| = \\
&|\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 \wedge k = k_1] \Pr[k = k_1]| + \\
&|\Pr[\mathcal{A}(G_{\mathcal{F}'}(\lambda)) = 1 \wedge k \neq k_1] \Pr[k \neq k_1] - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]| \leq \\
&\left|1 \cdot 2^{-\lambda}\right| + \left|\Pr[\mathcal{A}(G_{\mathcal{F}}(\lambda)) = 1] \frac{2^\lambda - 1}{2^\lambda} - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]\right| \leq \\
&2^{-\lambda} + |\Pr[\mathcal{A}(G_{\mathcal{F}}(\lambda)) = 1] - \Pr[\mathcal{A}(G_{rnd}(\lambda)) = 1]|
\end{aligned}$$

We first split the probability in the PRF game using conditional probabilities. The inequality then follows from the triangular inequality of the absolute value. Then we apply the definition of conditional probabilities. For the following step, we assume that the adversary can break with probability 1 the PRF when it's the identity function, and notice that $\mathcal{F}'$ behaves just like $\mathcal{F}$ when $k \neq k_1$. For the last inequality, we rely on $\frac{2^\lambda - 1}{2^\lambda} \leq 1$. What we obtain in the end is the sum of two negligibles (by assumption that $\mathcal{F}$ is a PRF), thus $\mathcal{F}'$ is a PRF.

Knowing this, the proposed PRF can be broken easily when it's built from $\mathcal{F}'$. Assume the adversary is playing against an unknown function $f$, which could be the proposed PRF with key $k$ or a random function. The adversary can give $k_1$ as input, obtaining a certain $y$ as output, and then check wether the following queries $x_1, \ldots, x_n$ give as output $F_{x_1}(y), \ldots, F_{x_n}(y)$. If that is not the case, then with probability 1 the adversary is dealing with a random function. Otherwise, with probability almost 1 the adversary is dealing with the proposed PRF.

The proposed construction could still work in some other cases, but it's not general.

c. This is a PRF. Assume a distinguisher $\mathcal{D}$ exists for $F'_k$, we can then build a distinguisher $\mathcal{D}'$ for $F_k$. $\mathcal{D}'$ can query an unknown function $f$. It asks $\mathcal{D}$ for inputs $x_1, \ldots, x_n$, and for each $x_i$ gives to $\mathcal{D}$

$$y = f(x_i || 0) || f(x_i || 1)$$

When $\mathcal{D}$ outputs a decision, $\mathcal{D}'$ outputs the same.