Homework 3—Solutions

1. We define a one-to-one function $\text{pad} : \bigcup_{i=0}^{\ell} \{0,1\}^\ell \rightarrow \{0,1\}^{\ell'}$ that maps strings of length at most $\ell$ to strings of length exactly $\ell' = \ell + O(\log \ell)$. Then encrypt $m$ using key $k$ by computing $\text{Enc}_k(\text{pad}(m))$, where $\text{Enc}$ is any CPA-secure encryption scheme. Since $\text{pad}$ is one-to-one, the receiver will be able to decrypt and recover $m$.

There are many ways to define $\text{pad}$, but here is one:

$$\text{pad}(m) = \text{len}(m || m || 0^{\ell - |m|}),$$

where $\text{len}(m)$ is the length of $m$ written as an integer using exactly $\lfloor \log \ell \rfloor + 1$ bits. **Challenge**: Can you find a way to do this with $\ell' = \ell + 1$?

2. (a) $G'$ is a pseudorandom generator, since it runs $G$ on a uniform seed.

(b) $G'$ is not necessarily a pseudorandom generator. To see this, let $H : \{0,1\}^n \rightarrow \{0,1\}^{4n}$ be a pseudorandom generator, and define $G(s) = H(s_1 \cdots s_{n/2})$. As in the previous part, $G$ is a pseudorandom generator. But then

$$G'(s) = G(0^{|s|} || s) = H(0^{|s|}),$$

and clearly $G'$ is not a pseudorandom generator.

Fundamentally, the problem here is that $G'$ runs $G$ on an input that is not uniformly distributed.

(c) $G'$ is not necessarily a pseudorandom generator. To see this, let $H : \{0,1\}^n \rightarrow \{0,1\}^{2n}$ be a pseudorandom generator and define $G(s) = H(s_1 \cdots s_{n-1})$. As in the previous parts, $G$ is a pseudorandom generator. But then if the last bit of $s$ is 0 we have

$$G'(s) = G(s) || G(s + 1) = H(s_1 \cdots s_{n-1}) || H(s_1 \cdots s_{n-1})$$

(because when the last bit of $s$ is 0 then $s$ and $s + 1$ differ only in their final bit), and so with probability 1/2 the two halves of the output of $G'$ are the same; this is clearly not a pseudorandom generator.

Fundamentally, the problem here is that $G'$ runs $G$ on two correlated (rather than independent) inputs.

3. Define keyed function $F : \{0,1\}^n \times \{0,1\}^{\log n} \rightarrow \{0,1\}$ as follows: $F_k(i)$ outputs the $i$th bit of $k$, where the input $i$ is interpreted as an integer in the range $\{0,\ldots, n - 1\}$ and the bits of $k$ are numbered starting at 0. Note that $F$ is exactly implementing a lookup table based on the key $k$, and so $F_k$ for uniform $k$ is exactly a random function mapping $\log n$-bit inputs to 1-bit outputs. I.e., $F$ is a random function, which is stronger than being pseudorandom.
4. $F'$ is not a pseudorandom function. To see this, consider querying on the two inputs $0^{n-1}$ and $0^{n-2}1$. We have

$$F'_k(0^{n-1}) = F_k(0^n) || F_k(0^{n-1}1)$$

and

$$F'_k(0^{n-2}1) = F_k(0^{n-1}1) || F_k(0^{n-2}1^2);$$

note that the second half of $F'_k(0^{n-1})$ is equal to the first half of $F'_k(0^{n-2}1)$.

Formally, define the following attacker $A$ given $1^n$ and access to some function $g$:

- Query $y_0 = g(0^{n-1})$ and $y_1 = g(0^{n-2}1)$.
- Output 1 if and only if the second half of $y_0$ is equal to the first half of $y_1$.

As shown above, we have $\Pr_{k \leftarrow \{0,1\}^n}[A_{F'k}(1^n) = 1] = 1$. But when $g$ is a random function then $y_0$ and $y_1$ are independent, uniform strings of length $2n$, and so the probability that the second half of $y_0$ is equal to the first half of $y_1$ is exactly $2^{-n}$. Thus, $\Pr_{f \leftarrow \mathsf{Func}}[A_{f}(1^n) = 1] = 2^{-n}$, and the difference

$$\left| \Pr_{k \leftarrow \{0,1\}^n}[A_{F'k}(1^n) = 1] - \Pr_{f \leftarrow \mathsf{Func}}[A_{f}(1^n) = 1] \right|$$

is not negligible.

5. (a) This is not even EAV-secure: given the ciphertext $(r, c)$, an attacker can recover $m$ by computing $m = G(r) \oplus c$. (Note in particular that encryption does not use the key!)

(b) Because $F$ is a pseudorandom function, $F_k(0^n)$ is a pseudorandom value and this encryption scheme is analogous to the pseudo-OTP. So it is EAV-secure. But just like the pseudo-OTP, this scheme is deterministic and so cannot be CPA-secure.

(c) This scheme is similar to the scheme we covered in class, and even when multiple messages are encrypted all the inputs to $F_k$ will be distinct with overwhelming probability. Thus, this scheme is CPA-secure (and hence also EAV-secure).