Problem 1: One-Time Pad Cryptosystem

a. Proof. The NSA cipher cryptosystem is Shannon secure. Let $m_0 \in M$, and $c$ be any ciphertext. By the given condition, we know that for any $m_1 \in M$, $\Pr[c|m_0] = \Pr[c|m_1]$. Therefore, we obtain:

$$ \Pr[c] = \sum_{m_1 \in M} \Pr[m_1 \land c] = \sum_{m_1 \in M} \Pr[m_1] \Pr[c|m_1] = \sum_{m_1 \in M} \Pr[m_1] \Pr[c|m_0] = \Pr[c|m_0] \sum_{m_1 \in M} \Pr[m_1] = \Pr[c|m_0] $$

By Bayes’ rule, $\Pr[m_0] \Pr[c|m_0] = \Pr[c] \Pr[m_0|c]$, and thus by cancelling we have $\Pr[m_0] = \Pr[m_0|c]$.

b. No, the new system is not Shannon secure. Since the key cannot be all zeros, the ciphertext and the message cannot be the same. Therefore, for $m \in \{0, 1\}^n$, if $c = m$, then $\Pr[m] = \frac{1}{2^n}$, but $\Pr[m|c] = 0$.

c. First note that,

$$ \Pr[k \leftarrow K_{0,n} : k = 0^n] = \frac{1}{2^n} \\
\Pr[k \leftarrow K_{1,n} : k = 0^n] = 0. $$

This means that if $k = 0^n$, then $k$ must have come from $K_{0,n}$. Moreover, for any fixed $k \in \{0, 1\}^n$ with $k' \neq 0^n$,

$$ \Pr[k' \leftarrow K_{0,n} : k = k'] = \frac{1}{2^n} \\
\Pr[k' \leftarrow K_{1,n} : k = k'] = \frac{1}{2^n - 1}. $$
Therefore,

\[
\Pr[b \leftarrow \{0, 1\}; \ k' \leftarrow K_{b,n} : b = 1 \mid k' = k] = \frac{\Pr[b \leftarrow \{0, 1\}; \ k' \leftarrow K_{b,n} : k = k' \land b = 1]}{\Pr[b \leftarrow \{0, 1\}; \ k' \leftarrow K_{b,n} : k = k']}
\]

\[
= \frac{\frac{1}{2} \cdot \frac{1}{2^{n-1}}}{\frac{1}{2} \left(\frac{1}{2^n}\right) + \frac{1}{2} \left(\frac{1}{2^n-1}\right)}
\]

\[
= \frac{1}{2^{n-1} + 1}
\]

\[
= \frac{1}{2^{n+1} - 1}
\]

\[
= \frac{1}{2} + \frac{1}{2n+2 - 2}.
\]

Thus, for any \(k \in \{0, 1\}^n\) with \(k \neq 0^n\), it is more likely that \(k \leftarrow K_{1,n}\). Hence, the best possible distinguisher will output 0 if \(k = 0^n\) and 1 otherwise (since this gives the highest probability of correctness on each input).

If \(A\) is the distinguisher described above,

\[
c_A(n) = \frac{1}{2} \Pr[k \leftarrow K_{0,n}; \ b' \leftarrow A(k, 1^n) : b' = 0] + \frac{1}{2} \Pr[k \leftarrow K_{1,n}; \ b' \leftarrow A(k, 1^n) : b' = 1]
\]

\[
= \frac{1}{2} \Pr[k \leftarrow K_{0,n} : k = 0^n] + \frac{1}{2}
\]

\[
= \frac{1}{2} \left(\frac{1}{2^n} + 1\right)
\]

\[
= \frac{1}{2} + \frac{1}{2^n+1}.
\]

Since \(\nu(n) = \frac{1}{2^n+1}\) is negligible, \(K_{0,n}\) and \(K_{1,n}\) are statistically indistinguishable.

d. Assume \(A\) is an algorithm that distinguishes between \(C_{0,n}(m)\) and \(C_{1,n}(m)\) with non-negligible advantage \(\epsilon\). Consider the following algorithm \(B\) to distinguish between \(K_{0,n}\) and \(K_{1,n}\): on input \(k\), compute \(c = k \oplus m\), run \(A(c, 1^n)\), and output the result.

If \(k \leftarrow K_{0,n}\), then \(c = k \oplus m = E(m, k)\), so this is precisely the distribution \(C_{0,n}(m)\). Similarly for \(K_{1,n}\) and \(C_{1,n}(m)\). Therefore, \(B\) will be correct precisely when \(A\) is correct. Since \(A\) distinguishes with advantage \(\epsilon\), so does \(B\). However, by part (b) we know that \(B\) must have negligible advantage. This contradicts the assumption that \(\epsilon\) was non-negligible, and therefore \(A\) cannot exist. Hence \(C_{0,n}(m)\) is statistically indistinguishable from \(C_{1,n}(m)\) for any \(m\).

**Problem 2: Negligible functions**

a. A function like \(\nu(k) = 2^{-k}\) is certainly negligible, and also always positive.

b. 1) This is not negligible. Consider \(p(k) = 1\) as a counterexample.

2) This is not negligible. Consider \(p(k) = 1\) as a counterexample.
3) This is tricky, but not negligible. Consider \( p(k) = k \) and \( \nu_i(k) = k \max(0, i + 1 - k) + \nu(k), \) for some negligible \( \nu, \) as a counterexample. \( \nu_i \) is negligible because, for all \( k > k_0 = i + 1, \) \( \nu_i(k) = \nu(k). \) However, the sum \( \sum_{i=1}^{k} \nu_i(k) \) simplifies to \( k + k\nu(k), \) which cannot be negligible.

4) Yes, this is negligible.

5) This is not negligible. Consider \( \nu(k) = 2^{-k} \) and \( p(k) = k \) as a counterexample.

6) Yes, this is negligible.

7) This is not negligible. Consider \( p(k) = k \) as a counterexample.

8) This is not negligible. Consider \( \nu(k) = 2^{-k} \) and \( c = 1 \) as a counterexample.

c. This does not follow. We can take a function like \( \epsilon(k) = \begin{cases} \text{negligible} & \text{for } k \text{ even} \\ 1 & \text{for } k \text{ odd} \end{cases} \) as a counterexample.

### Problem 3: Notation Practice

a. This can be written \( \Pr[m \leftarrow \{0,1\}^k; r \leftarrow \{0,1\}^k; c = m \oplus r : c = \check{c}]. \)

b. We denote \( S \) as our sample space of questions, and \( \text{Alg} \) as the algorithm from homework 1.

\[
\begin{align*}
\Pr \text{[success]} &= \Pr[q \leftarrow S; a \leftarrow \text{Alice}(q) : a \text{ is correct}] \\
&= \Pr[a \leftarrow \text{Alice}(q) : a = T \text{ if } q \text{ is } T] \times \Pr[q \leftarrow S : q \text{ is } T] \\
&+ \Pr[a \leftarrow \text{Alice}(q) : a = F \text{ if } q \text{ is } F] \times \Pr[q \leftarrow S : q \text{ is } F] \\
&= \Pr[a_A \leftarrow \text{Alice}(q); a_B \leftarrow \text{Bob}(q) : a_A = a_B = T \text{ if } q \text{ is } T] \times \Pr[q \leftarrow S : q \text{ is } T] \\
&+ \Pr[a_A \leftarrow \text{Alice}(q); a_B \leftarrow \text{Bob}(q) : a_A = F \text{ or } a_B = F \text{ if } q \text{ is } F] \times \Pr[q \leftarrow S : q \text{ is } F] \\
&= \Pr[q \leftarrow S : q \text{ is } T] \\
&\times (\Pr[a_A \leftarrow \text{Alice}(q) : a_A = T \text{ if } q \text{ is } T] \times \Pr[a_B \leftarrow \text{Bob}(q) : a_B = T \text{ if } q \text{ is } T]) \\
&+ \Pr[q \leftarrow S : q \text{ is } F] \\
&\times (\Pr[a_A \leftarrow \text{Alice}(q) : a_A = F \text{ if } q \text{ is } F] \times \Pr[a_B \leftarrow \text{Bob}(q) : a_B = T \text{ if } q \text{ is } F] \\
&+ \Pr[a_A \leftarrow \text{Alice}(q) : a_A = T \text{ if } q \text{ is } F] \times \Pr[a_B \leftarrow \text{Bob}(q) : a_B = F \text{ if } q \text{ is } F]) \\
&= (0.8 \times 0.15) \times 0.4 + (0.35 \times 0.15 + 0.65 \times 0.85 + 0.35 \times 0.85) \times 0.6 \\
&= 0.5895
\end{align*}
\]

### Problem 4: One Way Function: Definition

a. Let \( f \) be an arbitrary one-way function, and suppose for a contradiction that it is not computable-but-uninvertible (henceforth, “uninvertible”).

By the definition of uninvertible, we know there must then exist some probabilistic polynomial-time algorithm \( A \) such that, for all \( k \) and for all \( x \in \{0,1\}^k, \) on input \( (1^k, f(x)), \) \( A \) outputs \( x' \) such that \( f(x) = f(x'). \) This is the same as saying for all \( x, \) for all \( x \in \{0,1\}^k, \)

\[
\Pr[x' \leftarrow A(1^k, f(x)) : f(x) = f(x')] = 1
\]

This implies for all \( k, \)

\[
\Pr[x \leftarrow \{0,1\}^k; x' \leftarrow A(1^k, f(x)) : f(x) = f(x')] = 1
\]

HW 2 – Solutions-3
which contradicts the definition of one-way. Therefore if \( f \) is a one-way function, it must also be uninvertible.

b. There are two major errors in the proof. Below, the original proof is given again, with footnotes marking where the mistakes occurred.

We give a reduction: we show that an algorithm \( A \) that breaks the “one-wayness” of \( f \) also breaks that “uninvertibleness” of \( A \). Thus, the reduction accomplishes the contrapositive: not one-way implies not computable-but-uninvertible.

The reduction proceeds as follows: on input \( y = f(x) \), run \( A \), giving it input \( y \). With non-negligible probability, \( A \) outputs \( x' \) such that \( f(x') = y = f(x) \).\(^1\) If \( A \) outputs such \( x' \), output it. Else, run \( A \) again until it does.

Analysis of the reduction: Correctness follows because the reduction does not halt until it finds a correct \( x' \). Expected polynomial-time follows because \( A \) outputs a correct \( x' \) with non-negligible probability \( \epsilon(k) \), and \( \epsilon(k) \geq 1/p(k) \) for some polynomial \( p(k) \),\(^2\) so we need to run \( A \) \( 1/\epsilon(k) \leq p(k) \) times before it produces a correct \( x' \).

Therefore, if \( f \) is a computable-but-uninvertible function, then it is also a one-way function.

---

\(^1\)If a function is not one-way, then there exists a PPT \( A \) and a nonnegligible function \( \epsilon \) such that for all \( k \),

\[
\Pr[x \leftarrow \{0,1\}^k; x' \leftarrow A(1^k, f(x)) : f(x) = f(x')] = \epsilon(k)
\]

It does not follow that for all \( k \), for all \( x \),

\[
\Pr[x' \leftarrow A(1^k, f(x)) : f(x) = f(x')] = \epsilon(k)
\]

This means that when we give \( A \) a particular \( y \), it is not necessarily the case that \( A \) outputs \( x' \) such that \( f(x') = y = f(x) \) with non-negligible probability. In fact, there might be some \( y \) for which \( A \) never outputs a correct answer. In this case, the reduction will not even halt.

\(^2\)It is not the case that for any nonnegligible function \( \epsilon \), there exists a polynomial \( p \) such that for all \( k \), \( \epsilon(k) \geq 1/p(k) \).