Alex Berke March 19, 2023

6.5610 Problem Set 1

Collaborators: *Helen Lu, Ashley Chen*

Problem 1-1. Pseudorandom functions and one-way functions

(a) g is a PRF: The outputs of f $f(k_1, x), f(k_2, x)$ are pseudorandom. Even though they are from the same message, they are indistinguishable because they use separate keys and f is a PRF. Since the output of g is the concatenation of these two pseudorandom distributions, the output of g is also pseudorandom and therefore g is also a PRF.

(b) g is not a PRF. Attack: Use input (x_i, x_i). i.e. use input (x_1, x_2) s.t. $x_1 = x_2$; i.e. where first n bits equivalent to second n bits. A poly-time adversary can do this t times and observe that the output first m bits always equals the second m bits. i.e. g output is distinguishable from a random distribution.

(c) g is not a PRF. Attack: Like above, use input (x_i, x_i). Do this t times. Observe the output is always then 0. Clearly g is not random.

(d) g is a PRF: g essentially inputs two slightly different values $(x||0,\ x||1)$ to f and concatenates the outputs. Since f is a PRF, despite the similarity of the inputs, the outputs are pseudorandom. The concatenation of these output pseudorandom values is then also pseudorandom.

(e) g is a OWF: output of g is simply $f(x)$ padded with 0's. Since $f(x)$ is a OWF, then $g(f)$ must also be a OWF. i.e. since cannot invert f, then cannot invert g.

(f) g is not necessarily a OWF. We can have a OWF f s.t. f only operates on the 2nd $n/2$ bits. Such an f can be a OWF while g is then not.

(g)

(h) g is a OWF. In order to be a OWF, f must operate on more than just the 1st bit of x. Even though g reveals the 1st bit of x, since this information cannot be used to reverse $f(x)$, then $g(x)$ also cannot be reversed.

Problem 1-2. From functions to permutations

(a) D_f is not a PRP. Attacker can send input of the form (x_i, y_i) t many times and observe the first n bits in the output are always equivalent to y_i. Thus D_f is distinguishable from random.

(b) There is an attack to find D_f^2 is not pseudorandom and therefore not a PRP.

$D_f^2((k_1, k_2), (x, y)) = D_f(k_2, (D_f(k_1, (x, y)))) = D_f(k_2, (y, x \oplus f(k_1, y))) = (x \oplus f(k_1, y), y \oplus f(k_2, x \oplus f(k_1, y)))$

An attacker can use input (x_i, y) with varying x_i and consistent y. The intermediate value $f(k_1, y)$ is then consistent and the attacker can then use the first value in the output tuple, $x_i \oplus f(k_1, y)$ and apply $\oplus x_i$ and then observe that the value is always the same: $f(k_1, y) = x_i \oplus f(k_1, y) \oplus x_i$.

(c) Given z_i is uniformly randomly distributed in $\{0, 1\}^{2n}$, then $\{z_i, D_f^2((k_1, k_2), z_i)\}_{i \in [m]}$ and $\{z_i, u_i\}_{i \in [m]}$ are indistinguishable if $D_f^2((k_1, k_2), z_i)$ is indistinguishable from a uniformly random distribution in $\{0, 1\}^{2n}$. This is the case if $D_f^2((k_1, k_2), z_i)$ is pseudorandom, which it is: We can divide z_i into its (x, y) components and make the following argument.

Since f is a PRF, then $x \oplus f(k, y))$ is pseudorandom (same reason as for 1g). As above,

$D_f^2((k_1, k_2), (x, y)) = (x \oplus f(k_1, y), y \oplus f(k_2, x \oplus f(k_1, y))) = (a, b)$

Since $x \oplus f(k, y)$ is pseudorandom, and f uses independently random keys to output a and b, then a and b are also pseudorandom. And therefore their concatenation is pseudorandom.

The attack from 2b does not apply because z_i is uniformly randomly distributed versus constructed from a chosen (x, y).

Problem 1-3. Pseudorandom permutations

(a) Yes, f_F must produce an output for every input. Assume it did not. Then $\exists$ x s.t. $F(0||x) = F(1||y_1)$ and $F(1||y_1) = F(1||y_2)$ and $F(1||y_2) = F(1||y_3)$ and so on... In order to not end with output of the form $(0||y)$ then there must be a cycle s.t. input values to F with first bit 1 yield output with first bit 1. However, also there is also input $(0||x)$ s.t. first bit is 0 and output first bit is 1. Yet if both of these statements are true then F cannot be a permutation.

(b) If f_F is not a permutation then $\exists$ distinct x_1, x_2 s.t. $f_F(x_1) = f_F(x_2)$. This means $f_F(x_1) = F(0||x_1) = (0||y)$ and $f_F(x_2) = F(0||x_2) = (0||y)$ which implies F maps 2 distinct input to same output $(0||y)$. And therefore F is not a permutation – a contradiction.

(c) We can observe that any time the 1st bit of input for F is 0, then the 1st bit of output of F is also 0. For this reason F's output is distinguishable from a random distribution.

(d) It is possible to have a PRP F s.t. for 2 distinct x_1, x_2, $F(x_1) = y||0$ and $F(x_2) = y||1$. Then f'_F maps 2 distinct x_1, x_2 to same output (y) and f'_F is not a permutation.

Problem 1-4. Programming and substitution ciphers

(a)
- note I found 8 plausible keys that mapped bother ciphers to words in the dictionary. I chose the one that made the most likely sentences.
- c1: tomorrow snow is a welcome surprise
- c2: computer science will be the future
- code: `https://github.com/aberke/applied-crypto-and-security-6.5610/blob/master/pset1/programming/otp.py`

(b)
- my email is aberke@mit.edu
- encrypted ascii charcters: [58, 75, 90, 103, 85, 90, 124, 65, 88, 40, 92, 90, 72, 67]
- encrypted hex: 3a4b5a67555a7c4158285c5a4843
- code (jupyter notebook): `https://github.com/aberke/applied-crypto-and-security-6.5610/blob/master/pset1/programming/substitution-cipher.ipynb`