

Chapter 2

Section Example 2.3

se2_30ts

Homework 2 Problem 1

hw2_10ts

Consider a variant of one-time pad where we avoid choosing the all-zeroes key. The modified KeyGen algorithm can be written as:

KeyGen:
do
 $k \leftarrow \{0, 1\}^\lambda$
until $k \neq 0^\lambda$
return k

Hence k is uniformly distributed over the *set of all nonzero strings* of length λ . The Enc and Dec algorithms are the same as normal one-time pad. Formally show that this new encryption scheme does not satisfy one-time secrecy. Explicitly state the libraries that are relevant for this problem; write a calling program; derive the relevant output probabilities.

Chapter 5

Homework 5 Problem 1

hw5_1G

Let $G : \{0, 1\}^\lambda \Rightarrow \{0, 1\}^{3\lambda}$ be a secure length- **tripling** PRG. For each function below, state whether it is also a secure PRG. If the function is a secure PRG, give a proof. If not, then describe a successful distinguisher and explicitly compute its advantage.

- (a)

$$\begin{array}{l} \overline{H(s) :} \\ x := G(s) \\ y := G(0^\lambda) \\ \text{return } x||y \end{array}$$

(b)

$$\begin{array}{l} \overline{H(s) :} \\ x := G(s) \\ y := G(0^\lambda) \\ \text{return } x \oplus y \end{array}$$

(c)

$$\begin{array}{l} \overline{H(s) :} \\ x||y||z := G(s) \\ w := G(x) \\ \text{return } x||y||z||w \end{array}$$

Chapter 6

Homework 6 Problem 1

hw6_1Prg

Let F be a secure PRF with $in = out = \lambda$. Define the following function:

$$F'(k, m) = F(k, m) || F(k, F(k, m))$$

Show that F' is **not** a secure PRF.

Homework 6 Problem 2

hw6_2Prg

Show that a 2-round keyed Feistel cipher **cannot** be a secure PRP, no matter what its round functions are. Your attack should work without knowing the round keys, and it should work even with different (independent) round keys.

Hint: A successful attack requires two queries.

Chapter 7

Homework 7 Problem 2

hw7_2Cpa

Let F be a secure PRP with blocklength λ . Show that the following construction does not have CPA/CPA\$ security:

Enc(k, m):
 $s_1 \leftarrow \{0, 1\}^\lambda$
 $s_2 := s_1 \oplus m$
 $x := F(k, s_1)$
 $y := F(k, s_2)$
 return (x, y)