600.472 Theoretical Cryptography	February 25, 2008
Handout 5: Homework 3	
Instructor: Susan Hohenberger	*TA: Matthew Green*

Due at start of lecture on Thursday, March 6, 2008.

Problem 1 *Pumping Pseudorandomness (20 points)*

Suppose you are given a PRG G such that $|G(x)| = |x|+1$ and a polynomial p. Construct a G' such that $|G'(x)| = p(|x|)$, and prove that G' is also a PRG. A certain level of informality is acceptable here, so long as you are clear and hit the main proof ideas.

Problem 2 *On Pseudorandom Functions (30 points)*

Let $\{f_s : \{0,1\}^k \to \{0,1\}^k \mid s \in \{0,1\}^k\}$ be a family of pseudorandom functions. For each of the following, decide if the proposed construction is:

- *always* a PRF regardless of how f is implemented (provided that f is a PRF). In this case, prove that the construction is a PRF.
- *never* a PRF regardless of how f is implemented (provided that f is a PRF). In this case, give a generic attack for distinguishing.
- *might not* be a PRF depending on how f is implemented. In this case, give a counterexample of a specific PRF f[1] for which the resulting construction is not a PRF.

1. $F_s(x) = f_s(x)||f_s(\bar{x})$ (i.e., flip the bits of x in the second evaluation of f)
2. $G_s(x) = f_s(f_x(x))$
3. $H_s(x) = f_s(x+1)$
4. BONUS (10 additional points): $I_s(x) = f_s(x) \oplus s$

Problem 3 *Understanding CBC-mode Encryption (30 points)*

Let's further explore one of the different modes of encryption discussed in class.

1. (from Katz-Lindell 3.17) Present a formula for decryption of CBC-mode encryption. Can it be parallelized?
2. (from Katz-Lindell 3.22) Show that CBC mode of encryption does not yield CCA-secure encryption (regardless of F).
3. (Katz-Lindell 3.16) Consider a variant of CBC-mode encryption where the sender simply increments the IV by 1 each time a message is encrypted (rather than choosing the IV at random each time). Show that the resulting scheme is *not* CPA-secure.

[1] Build such a PRF generically assuming the existence of PRFs.

Problem 4 *Attacking Twisty Blockciphers (20 points)*

Recall the Twisty[2] construction of a pseudorandom permutation (blockcipher) from a pseudorandom function. The formula for this blockcipher is: $M = (L_0, R_0)$:

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= f_{i+1}(R_i) \oplus L_i \end{aligned}$$

where the output after n rounds is (L_n, R_n), and each f_i is a pseudorandom function specified by the key.

Definition 1 (Blockcipher) *A blockcipher* (Gen, F) *is* secure *if for all PPT distinguishers* D, *there exits a negligible function* ϵ *such that for a random key* $K \in \mathsf{Gen}(1^k)$,

$$|\Pr[\mathsf{D}^{F_K(\cdot)}(1^k) = 1] - \Pr[\mathsf{D}^{\Pi(\cdot)}(1^k) = 1]| \leq \epsilon(k)$$

where Π *is chosen uniformly at random from the set of permutations on* k*-bit random strings.*

Definition 2 (Strong Blockcipher) *A blockcipher* (Gen, F) *is* strongly secure *if for all PPT adversaries* D, *there exists a negligible function* ϵ *such that for a random key* $K \in \mathsf{Gen}(1^k)$,

$$|\Pr[\mathsf{D}^{F_K(\cdot), F_K^{-1}(\cdot)}(1^k) = 1] - \Pr[\mathsf{D}^{\Pi(\cdot), \Pi^{-1}(\cdot)}(1^k) = 1]| \leq \epsilon(k)$$

where Π, Π^{-1} *are inverses and* Π *is randomly chosen as above.*

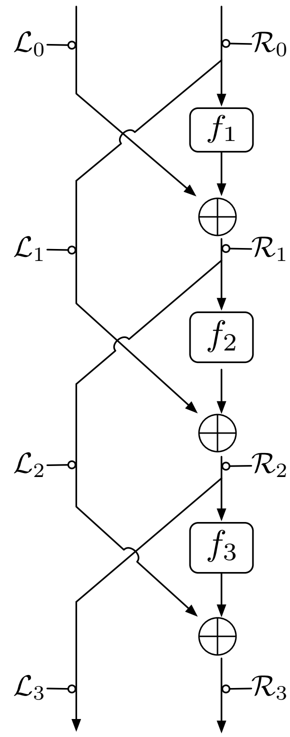

Figure 1: An illustration of 3 rounds of Twisty.

It is known that 3 rounds of Twisty forms a secure blockcipher and that 4 rounds of Twisty forms a strongly secure blockcipher. In this problem, you are asked to show that these formulations are round optimal by describing algorithms D that contradict the above definitions for fewer rounds. For example, 1 round is not a secure blockcipher because for input (L_0, R_0), D can call its oracle an obtain the output (X, Y). If $X = R_0$, then D outputs 1; otherwise, D outputs 0. If D's oracle is F_K, D will always output 1; however, if D's oracle is a random permutation Π than it will output 1 with probability $1/2^{|X|}$.

1. Show that 2 rounds of Twisty is *not* a secure blockcipher.
2. Show that 3 rounds of Twisty is *not* a strongly secure blockcipher.
   (HINT: there is a solution using only three oracle calls.)

[2] We'll call this blockcipher by its proper name in the solutions.