Homework 1

CSC 152/252 – Cryptography
Due: 11:59pm, Sunday, September 16

Read the document "Homework procedures" posted in Piazza resources for how to turn in your homework and policies on collaboration. A small sampling of your homework may be graded over the semester. To minimize the chance that bad luck causes you a bad grade, do every problem to the best of your ability. Show your work. Ask questions if you need help.

What to turn in: Program file `hw1.c` and a single file with your written solutions named `hw1.pdf`, `hw1.doc`, or `hw1.docx`. Misnamed files and files over 1MB will receive no credit.

Written Problems:

1) As you may have seen in CSC 28, if f is a binary relation between sets A and B with $|A|$ ordered pairs in the relation and every element of A occurs exactly once as a first element in an ordered pair, then the relation can be viewed as a function $f : A \to B$.

a) Let $A = \{1, 2, 3\}$ and $B = \{a, b, c\}$. Is $f = \{(1, a), (2, b)\}$ an invertible function? If so, what is its inverse (given as a set of ordered pairs). If not, explain in one short sentence.

b) Let $A = \{1, 2, 3\}$ and $B = \{a, b, c\}$. Is $f = \{(1, a), (2, b), (3, b)\}$ an invertible function? If so, what is its inverse (given as a set of ordered pairs). If not, explain in one short sentence.

c) Let $A = \{1, 2, 3\}$ and $B = \{a, b, c\}$. Is $f = \{(1, a), (2, b), (3, c)\}$ an invertible function? If so, what is its inverse (given as a set of ordered pairs). If not, explain in one short sentence.

2) Let $\mathbb{Z}_n$ be shorthand for the set containing the n smallest non-negative integers (eg, $\mathbb{Z}_3 = \{0, 1, 2\}$). Is $f : \mathbb{Z}_5 \to \mathbb{Z}_5$ defined as $f(x) = 2x \bmod 5$ an invertible function? If so, what is its inverse (given either as a set of ordered pairs or as a formula). If not, explain in one short sentence. *Note: since this signature is of the form $A \to A$, if it is invertible then it can also be called a permutation or permutation function.*

3) a) How many functions exist with signature $f : \mathbb{Z}_4 \to \mathbb{Z}_5$?

b) Given that a and b are positive integers, how many functions exist with signature $f : \mathbb{Z}_a \to \mathbb{Z}_b$?

a) How many permutation functions exist with signature $f : \mathbb{Z}_4 \to \mathbb{Z}_5$?

a) How many permutation functions exist with signature $f : \mathbb{Z}_4 \to \mathbb{Z}_4$?

b) Given that a and b are positive integers, how many permutation functions exist with signature $f : \mathbb{Z}_a \to \mathbb{Z}_b$?

4) Let `rand(n)` be a library function that evaluates to a random integer in $\mathbb{Z}_n$ each time it is called (like Java's `Random.nextInt(n)`). Write a method called `createRandomFunction` (right here in your written homework) in C or Java that takes a positive integer n as a parameter and returns an array with n elements each uniformly distributed in $\mathbb{Z}_n$. Essentially I'm asking you to write a method that specifies a random function $\mathbb{Z}_n \to \mathbb{Z}_n$ using the table filling method (ie, `a = createRandomFunction(10)` fills `a` with random values and then `a[0]` would tell you what 0 maps to, `a[1]` tells you what 1 maps to, etc.).

5) Do Problem 4 again, but this time name the method `createRandomPermutation` and make the array a permutation (ie, 0 through $n - 1$ each appear exactly once). For full credit, make your method run on $O(n)$ time.

Programming:

P1) We saw in class that one way to write an invertible function is to use a Feistel construction. Here's example pseudocode that uses x^2 as its mixing function.

```
unsigned perm(unsigned x) {
    hi = x >> 16;                    // hi initialized from high 16 bits of x
    lo = x & 0xFFFF;                 // lo initialized from low 16 bits of x
    hi = hi ^ (lo * lo);             // Feistel step: hi = hi xor f(lo)
    lo = lo ^ (hi * hi);             // Feistel step: lo = lo xor f(hi)
    x = (hi << 16) | (lo & 0xFFFF);  // Reform x from low 16 bits of hi and low 16 bits of lo
    return x;
}
```

Develop this pseudocode into a C function with the intended signature. Test it thoroughly by picking strategic inputs and comparing hand-calculated results with computer-generated results. Next, write an inverse function with the same signature but named `inverse_perm`. For every unsigned integer `x` it should be that `inverse_perm(perm(x))==x` .

Put your code into a file `hw1.c` and submit it via DBInbox. Your file should include only the two required functions and no `main`, should be appropriately documented, and should compile without warning or error when compiled using `gcc` or `clang` with compiler options `-std=c99 -W -Wall -Wpedantic -c`. The only headers your file is allowed to include are the standard ANSI C headers.