YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 367: Cryptography and Computer Security *Handout #8*
Professor M. J. Fischer *April 3, 2019*

Homework Assignment 7

Due on Thursday, April 11, 2019.

Extending Hash Functions

Happy threw together a hash function h : 32-bits → 16-bits, which he implemented by a C function `hash32()`. Adapting Method 2 from slide 22 of Lecture 15, Happy defined a new hash function H: 64-bits → 16-bits and implemented it by a C function `hash64`. Since he didn't know how to find colliding pairs for h, he thought that H would also be collision-free.

Clever Clem was able to find lots of colliding pairs for H. He didn't want to tell Happy how he did it, but he presented Happy with a file `H-collisions` of colliding pairs for H, each line of which consists of two 64-bit whitespace-separated hex numbers.

Your job is to write a program `breakHash.c` that applies the ideas presented on slide 23 of Lecture 15 to find corresponding colliding pairs for h. Your program should take the name of a file containing pairs of collisions for H as a command line argument. It should read each line, determine whether case 1 or case 2 applies, and find the corresponding colliding pair for h. You should then write a line to standard output consisting of 5 numbers: the original colliding pair for H, the case number that pertains (1 or 2), and the colliding pair for h described by that case. Colliding pairs should be written in hex with the `0x`-prefix (as in the input file). The case number should be written as a single digit. In case 2, if both $m_1 \neq m'_1$ and $m_2 \neq m'_2$, then print the *first* colliding pair for h.

Do *not* attempt to reverse-engineer `hash32()` (although this is possible to do with a little thought and cleverness). Rather, the goal of this problem is for you to apply the cited method from Lecture 15 for finding colliding pairs for `hash32()` given colliding pairs for `hash64()`. For this reason, I am only releasing the object code for `hash32()` and `hash64()`. Your program can call these functions, but I'm purposely not giving you the source code.

You will find the three files that you need for this assignment in Zoo directory `/c/cs367/assignments/hw7/`:

`hw7.h` is the header file for `hash32()` and `hash64()`. It also contains some useful typedef's and macros for dealing with bit strings represented by unsigned integers.

`libhw7.a` contains linux binaries for `hash32()` and `hash64()` so that you can compute them.

`H-collisions` is Clever Clem's file of colliding pairs.

In order to call `hash32()` and `hash64()` from your own program, `breakHash.c`, you will need to do three things:

(a) Put `hw7.h` and `libhw7.a` into your working directory along with your code.

(b) #include the header file `hw7.h` in your code.

(c) Link your compiled code to the library `libhw7.a`.

You can compile and link with the single command line:

```
gcc -o breakHash -std=c99 -Wall -O1 -g -L. breakHash.c -lhw7
```

The switch `-L.` says to search for the library in your working directory. The switch `-lhw7` says to link to the library `hw7`, which resides in the file `libhw7.a`.

The folder `mac-osx` contains a version of `libhw7.a` that you can use if you choose to develop your code on a Mac. However, you should compile and test your code on the Zoo before submitting.