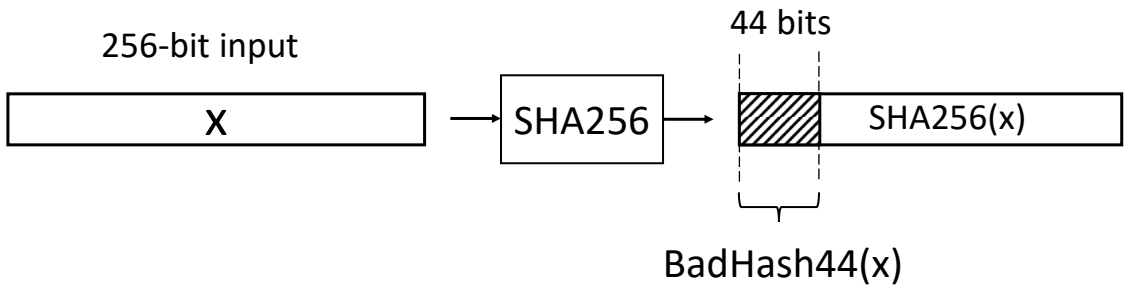
CSCE 4050/5050 Applications of Cryptography

Programming Project 2 – Birthday attack (contains 2 pages)

Due on April 7 at 11:59pm

Birthday attack is a generic algorithm for finding collisions in hash functions.

Project Task: Write a program that runs the birthday attack against a hash function defined next. This hash function is called "BadHash44" – it is an intentionally short hash function constructed using SHA256. Specifically, BadHash44(x) := SHA256(x)[1..44], i.e., it takes the first 44 bits of the output of SHA256. The construction of BadHash44 is illustrated in the following figure:



Your task is to find two (arbitrary) inputs to the function BadHash44 which result in the same output, i.e., to search for a collision in BadHash44. Your program must compute such the two inputs and the corresponding output for BadHash44; use the hex format when reporting all these values.

Requirement (IMPORTANT): Implement the generic birthday attack algorithm described in Lecture 6-1 (see also Sec. 8.3 of Boneh-Shoup's textbook). Your program will write the table of random messages m_i and their corresponding hash values t_i (use the hex format for these values) into a text file "hash.txt" or "hash.csv". NOTE: This table must be sorted on the hash values t_i. This file will be included into your submission package.

Tip: For certainty, use the input messages of size 256 bits. In principle, this can be any "large enough" value.

Feel free to use the programming language of your choice. Python is recommended.
In Python, the "hashlib" library contains an implementation of SHA256.

Organization: This project allows group work. The group set from Programming Project 1 will be used by default. One project report per group must be submitted by <u>any</u> of the group members.
<u>All group members will receive the same grade.</u>

Note: If you would like to change your group membership (join/leave/form a new team), you must notify the instructor by April 3 (Mon), 23:59pm.

Submission requirements:

- The project report must contain a short description of the task (collision search using the birthday attack), a short description of your program, and screenshots demonstrating that your program works according to the project task.
  (The size of your report will be between 2 and 7 pages.)
- The source code, the table of inputs/outputs of the hash function—a text file named "hash.txt" or "hash.csv", which contains values in the hex format, the executable(s) if any, and all other necessary files must be enclosed as a ZIP archive.
  NOTE: The source code must be properly commented.
- If you use Python, follow the guidelines listed below:
  - Use Python 3 (any version newer than 3.0).
  - Use SHA256 from "hashlib" library.
  - The hash input/output list must be saved in the file "hash.txt" or "hash.csv" located in the same folder as your code.
- If you use a language different from Python, your report should contain detailed instructions on how to run the code.

Remarks

1. Failure to provide components listed in the above submission requirements will result in reduced grade.
2. It is students' responsibility to demonstrate that the implementation is working.
   In cases when the code cannot be run, the screenshots placed in the report and the comments placed in the code will serve as evidence that the code is working (and that it is authentic).

Rubric:

- The code is worth 80 points, the breakdown is as follows:
  - The code is working: 70 points.
    - Either the grader is able to run it, or the grader is convinced via inspecting the code and the screenshots provided in the report.
    - Collision is found: 30 points. The remaining 40 points reflect how well the birthday attack algorithm is implemented.
  - The code is properly commented: 10 points.
- The project report is worth 20 points.
  - An adequate description of the task (collision search using the birthday attack) must be provided.