Applied Cryptography

MD5 Collisions

Luca Campa

Department of Computer Science
Universität Innsbruck

03 April, 2025

Overview

- Install `libboost-all-dev` (e.g. `sudo apt install libboost-all-dev`)
- Clone the following repository: https://github.com/brimstone/fastcoll
- Enter the `fastcoll` folder and modify the file `main.cpp` or `main.hpp`:
    - Add the following line before everything: `#define BOOST_TIMER_ENABLE_DEPRECATED 1`
- Run the following command: `g++-11 -O3 *.cpp -lboost_filesystem -lboost_program_options -lboost_system -o fastcoll -static && strip fastcoll`

Generate the collision

- Generate a file (let us call it `prefix.txt`) and add to it a small text like `helloworld!`
- Run the following command `./fastcoll -p prefix.txt -o file1.bin file2.bin`
- Run the following command `md5sum file1.bin file2.bin`
- How is the output?

Looking inside the files

- Run the following command `xxd file1.bin`
- Run the following command `xxd file2.bin`
- Why do you think there are so many zeros? What is the MD5 block size?
- How many bytes are different between the two files?

What happens if

- the chosen prefix is composed by 64 bytes?
- the chosen prefix is composed by 63 bytes?
- Use `xxd` or `hexdump` to view the bytes.

Why the prefix?

- The main reason is that we may desire to keep part of the file always the same. For instance:
  - the header of the files (which are essential to make the file readable as intended), e.g. given a PNG image, we can modify certain part and obtain another image with the same MD5 hash and still readable by image viewers.
  - the certificate fields, necessary if we want to impersonate another entity (theoretically speaking. . . don't do that).