Sean Evans
CS 6058
Data / Security and Privacy
Spring 2018
Project 4

# Project 4: Proof of Work

## Description

The proof of work tool has three major functions, including target file generation, solution generation, and solution verification.

The target file generation function produces a 256-bit binary target file using a difficulty factor between 0 and 256. Using the difficulty factor, the output file is generated with a matching number of leading zero bits, with the remaining trailing bits set to one. This target is used a large integer value by which solutions are evaluated.

The solution file generation function produces a candidate solution by generating a series of random bytes, which are appended to an input message and hashed using SHA256 and compared with the specified target value, until a solution is found which satisfies the target.

The solution verification function ingests a target, a message, and a candidate solution, and attempts to verify the candidate solution using the same concatenation, SHA256 hash, and comparison operation as in the solution generation function as described above.

## Implementation Details

The proof of work tool was implemented using C++, and uses the C++ Standard Library and Standard Template Library (STL), OpenSSL, and Boost Libraries.

Boost lexical cast was used to perform a conversion from an input string on the CLI to a native integer type.

The OpenSSL library was used to perform the SHA256 hashing operation.

The solution generation functionality was implemented using the C++ Standard Library's random number generator device and uniform integer distribution filter to generate a sequence of random bytes.

The solution evaluation is performed using the lexicographical compare function in the C++ STL, which performs byte-wise less-than comparison on arbitrary length byte strings.

The build system for the tool was implemented using CMake.

The tool was built and tested on a Windows 10 x64 system in the Cygwin x64 environment.

## Running Time Of Solution Generation

| difficulty | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|
| iterations | 10 | 10 | 10 | 10 | 10 | 10 |
| min run time | 1552 ms | 1255 ms | 1969 ms | 3401 ms | 147240 ms | 65977 ms |
| max run time | 65950 ms | 83319 ms | 222784 ms | 337083 ms | 811434 ms | 2486194 ms |
| mean run time | 25265 ms | 31916 ms | 70585 ms | 113663 ms | 360175 ms | 1116859 ms |
| median run time | 24770 ms | 31892 ms | 59102 ms | 109873 ms | 389793 ms | 1318544 ms |
| total run time | 252651 ms | 319166 ms | 705854 ms | 1136637 ms | 3601757 ms | 11168598 ms |

| difficulty | solution |
|---|---|
| 21 | ef 0d 40 f7 6f e8 a1 ab 3f 61 |
| 22 | 83 15 66 44 c8 c1 20 3c 02 7f |
| 23 | e9 7d aa 8a 54 25 3d 07 93 55 |
| 24 | b3 f8 b0 da 88 41 2e 8b 9a 34 |
| 25 | d0 ca 6c d9 40 89 82 72 e9 8a |
| 26 | 3e 97 ee ce 3a 59 ee 0a 99 a2 |