San José State University
CS266: Topics in Information Security
Autumn 2019

Homework 1

The goal of this homework is to build a basic blockchain with a proof-of-work mechanism and Merkle tree-based data storage.

We provide two Python modules that you may wish to use. The first one is `DataSimulator`, which simulates an I/O interface. Each time the function `getNewData`() is called, a set of *(publicKey, signature, string)* tuple is returned. (The actual data can be accessed by reading a JSON file, if you don't want to use Python)

```
from DataSimulator import DataSimulator
DSim = DataSimulator()

d = getNewData()
```

The second one is a naive implementation of elliptic curve cryptography functions. You will need signature verification:

```
import ECC

isVerified = ECC.verify(publicKey, message, signature)
```

Deliverables

1) Code that

   A) implements a blockchain—in particular, a main loop that repeatedly obtains new data and computes and outputs the hash $H(d)$ of data structure *d* (see instructions below).

   B) The data structure *d* must contain
   - a hash of the previous block
   - a Merkle Tree (see instruction D below)
   - a nonce

   C) The first 16 (binary) digits of each block hash must be 0.

D) the Merkle Tree must be build from all valid elements received from the DataSimulator, i.e. all valid *(publicKey, message, signature)* tuples where the signature is valid

2) Provide a Merkle Tree-proof that a specific item is part of the Merkle Tree in an ancestor of the last block. In detail, given the created blockchain after 5 blocks, show proof that the headline

```
cabinet meets to balance budget priorities
```

was “put on the blockchain.”

Evaluation

1a) 1 Point for running code that regularly receives the data and builds a correct hash chain
1b) 2 Points iff all 3 conditions are fulfilled
1c) 2 Points if the nonce is found correctly so that this condition is fulfilled
1d) 1 Point if the Merkle Tree is built correctly
1 Point if the correct elements are used to build the Merkle Tree

2) 2 Points if a correct proof of the given message in a Merkle Tree is provided
1 Point if a correct proof is provided of that Merkle Tree being part of the blockchain

Total: 10 Points