**Laboratory 10: Implementation of a Verified Merkle Root**

In this lab, you will be working with Java security package to construct a Merkle Hash Tree (MHT) of students grades that will later be used to calculate a merkle root. Then, this root value will be signed by an authentic person (let’s say by your instructor). Finally, this signed value will be verified to make sure that your friend is not making a prank on you (though it’s well past of 1st April, you never know!).

**Related resources:**

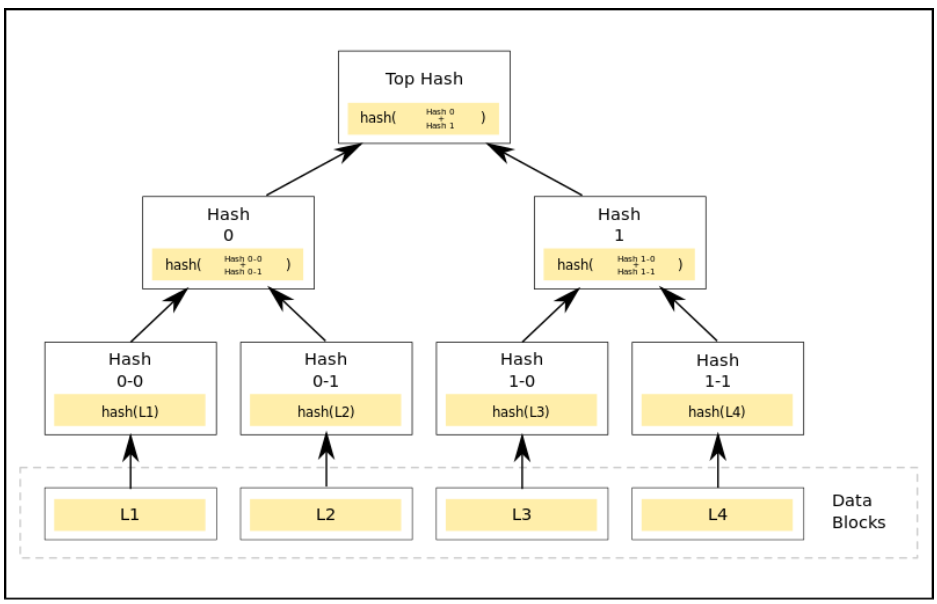
<https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html>

<https://docs.oracle.com/javase/tutorial/security/apisign/index.html>

**Objectives:**

* Hashing
* Merkle Trees
* Merkle Root
* Digital Signature

**Construction of Merkle Hash Tree**



**Part I: Calculate the merkle root**

1. Open the given input csv file (with 9 test grades) and store its content in an ArrayList.
2. Calculate list of SHA-256 hashes of original data (e.g. students marks given in a csv file)
3. If the list is empty, return a 32-byte (256 bits) root hash of all zeroes; otherwise
4. While the list contains two or more items:
   * If there are an odd number of nodes on any level of the merkle tree, the last node is duplicated and hashed with itself.
   * Combine each pair of adjacent entries with the SHA-256 hash of the two entries concatenated together.
5. Return the final remaining item in the list as the Merkle root.

**Part II: Get the merkle roots signed (by an authentic person/a malicious person)**

Create two pairs of keys (i.e. two set of public/private key). One key pair is intended for the instructor while the other one is meant for your friend whom you believe to be willing to tamper your mark to give a prank. The instructor will sign his calculated merkle root with his private key. Your friend will now tamper one of the 9 test grades, and calculate his own merkle root by following the steps mentioned above. He (your friend) will then sign it with his private key.

**Part III: Validate both of the calculated merkle roots**

You already know the public key of your instructor because he/she made it available on his/her lecture. Using that piece of information, you will need to verify whether both roots were signed by your instructor. Cleary, you are supposed to have one merkle root (that was originally signed by your instructor) verified. The other merkle root should not pass your verification.

**Part IV: Requirements**

Your program must use the following Security classes:

* java.security.KeyPair
* java.security.KeyPairGenerator
* java.security.PrivateKey
* java.security.PublicKey
* java.security.SecureRandom
* java.security.Signature
* java.security.SignatureException
* import java.security.MessageDigest
* import java.security.NoSuchAlgorithmException

**Sample Files**

* DigitalSignature.java
* SHA256.java
* input.csv