CS5430: System Security (Fall 2023) Programming Project

Phase 0: An insecure key-value store

General Instructions. Work together in a group of 2 or 3 students from the class. Group-forming assistance will be provided for those seeking help in forming a group.

This programming assignment should be implemented in Java version 11, which is available on the `ugclinux` computers.

Due Date. September 18, 2023 at 11:59 pm (in CMS)

The Project. This semester, CS5430 students will implement a key-value store that enforces certain security policies on access to the information it holds. The implementation will involve various pieces, each created as a separate phase of the project. The sequence in which these phases are assigned will parallel the order in which relevant topics are covered in lecture.

A key-value store is a database that stores pairs comprising a key and a value. The value associated with a given key can be retrieved by issuing a read operation that has that key as an argument; the value associated with a given key can be updated to a new value by issuing a write operation that has two arguments: a key and a new value.

The `Phase0` key-value store we are building actually will store two values for each key. One value is some data of type `V` (think: data), the other value is data of type `M` (think: metadata). Having the two values stored for each key allows the key-value store to maintain information about a value or about a key by using the metadata to hold that additional information. Later project phases will make use of this facility.

The key-value store that you build should be a Java object named `Phase0Impl` that constitutes an implementation of the `Phase0` Java interface. That interface can be downloaded from the CMS entry for phase 0. It is also given below including the comments that explain the intended effect of each operation. Notice that type variables `K`, `V,` and `M` are arguments to `Phase0` interface; we will instantiate `K`, `V,` and `M` when testing your system.

Once you have designed, programmed, and tested your system, you should submit to CMS the following files.

- `readme.txt` that gives a script that the grader can follow to compile and build your implementation for this interface. The result of this script should be a single `JAR` file named `Phase0.jar`. We will use this for testing your system.
- source code for a Java object named `Phase0Impl` that is your implementation of the Phase0 interface
- source code for any other Java objects that you have written for your implementation

```
import java.lang.IllegalArgumentException;
import java.util.NoSuchElementException;

public interface Phase0<K, V, M> {

  public void create(K key, V initVal, M initMetaVal);
// Thereafter key exists until key is deleted.  Associated with key will
be initVal and initMetaVal.

  public void writeVal(K key, V newVal) throws IllegalArgumentException;
// Makes newVal the value for key.  If writeVal is invoked when key does
not already exist then throw IllegalArgmentException.

  public void writeMetaVal(K key, M newMetaVal) throws
IllegalArgumentException;
// Makes newMetaVal the metavalue for key.  If writeMetaVal is invoked
when key does not already exist then throw IllegalArgmentException.

  public V readVal(K key) throws NoSuchElementException;
// Returns the value being stored for key.  If readVal is invoked when key
does not exist then throw NoSuchElementException.


  public M readMetaVal(K key) throws NoSuchElementException;
// Returns the metavalue being stored for key.  If readMetaVal is invoked
when key does not exist then throw NoSuchElementException.

  public void delete(K key);
// If key does exist then remove key, its value, and its metavalue from
the information being stored by the key/value store. If key does not exist
then take no action.
}
```