Hacettepe University, Department of Computer Engineering
BBM465 Information Security Laboratory
Experiment 3

Subject	:	Integrity Checker
Language	:	Java
Due Date	:	09/12/2020

1 Experiment

You are expected to develop a directory integrity checking program. The program monitors the changes on a specified directory using hash functions and digital signatures. The program must keep track of all changes, such as deleting, altering, and creating files in the monitored directory and record all these changes to a log file. The program should have three main functionalities: 1) create public/private key pairs and public key certificate 2) create a registry for a directory 3) check changes on the directory using registry file. The program must be developed as a console application and names as "*ichecker"*.

Creating Public/Private Key Pair:

Before starting monitoring operations, the program must be run to create public/private keys and associated public key certificate. The program should be run as follows:

ichecker createCert -k PriKey -c PubKeyCertificate

* *-k* specifies the path of the private key file, which is encrypted with AES algoritm.
* *-c* specifies the path of the public key certificate file

When this command is executed, *ichecker* will use *keytool* [1] to create a random public and private key pair for 2048 bit RSA algorithm. It also asks user to enter a password, which is be used to encrypt the private key file. When the user enters a password, MD5 hash operation is applied on the password and 128 bit output of MD5 function is be used as an AES key to encrypt private key file, which is specified with *–k* parameter. The private key file must contain some additional meaningful text data (such as, "This is the private key file") to understand if the user enters wrong password in the later operations.

Additionally, using *keytool*, the program will create a X.509 certificate file, which is specified with *–c* parameter. The certificate file contains the public key and must be signed with the private key (public and private key are the keys created by *keytool*). In other words, you need to create a self-signed certificate. This certificate and encrypted private key file will be used in the later operations.

Creating Registry File:

To start monitoring a directory, the program must be run to create a registry for the monitored directory. The program should be run as follows:

ichecker createReg -r RegFile -p Path -l LogFile -h Hash -k PriKey

* *-r* specifies the path of the registry file.
* *-p* specifies the path of the directory to be monitored by the program.
* *-l* specifies the path of the log file.
* *-h* specifies a hash function, which can be "MD5" or "SHA-256".
* *-k* specifies the path of the private key file, which is encrypted with AES algoritm.

After running above command, the program asks user for a password. When the user enters a password, AES key is obtained by applying MD5 hash operation on the password. Then, the program uses AES key to decrypt the private key file, which is specified by *–k* parameter. If the user enter a wrong password, the program must understand it by checking decrypted content of the private key file and terminate itself. This wrong password attempt must be logged to the log file, which is specified by *–l* parameter. The log format should be like this:

```
[time_stamp]: Wrong password attempt!
```

The format of the ***time_stamp*** should be in `time stamp = dd-MM-yyyy HH:mm:ss` format. Then the program creates the registry file specified by *–r* parameter. The registry file stores the path of all files in the monitored directory specified by *-p* parameter and hash values of file contents, which is calculated by using the hash algorithm specified by *-h* parameter. The format of registry file is as follows:

```
[Path_of_file1] H(file1)
[Path_of_file2] H(file2)
[Path_of_file3] H(file3)
. . .
#signature#
```

In this registry file, `[Path_of_file]` refers the full path of a file and `H(file)` refers to hash digest of the file. The last line of the registry file contains a signature, which is obtained by calculating the hash value of all content of the registry file except the last line and then by signing the hash value with the private key specified by *-k* parameter.

The program must also record the whole operation to the log file, specified with *– l* parameter. A sample for the log file after executing the above command should be like this:

```
[time_stamp]: Registry file is created at [Path_of_RegFile]!
[time_stamp]: [Path_of_file1] is added to registry.
[time_stamp]: [Path_of_file2] is added to registry.
[time_stamp]: [Path_of_file3] is added to registry.
. . .

[time_stamp]: 16 files are added to the registry and registry
creation is finished!
```

Checking Integrity:

After creating a registry file, the integrity of directory can be checked by using *ichecker*. To do that, the program should be run as follows:

ichecker check -r RegFile -p Path -l LogFile -h Hash -c PubKeyCertificate

In this command, *-c* specifies the path of the public key certificate file. When this program is executed, the registry file must be verified first by checking the signature located at the end of the file. For signature verification, the public key in the certificate must be used. If the verification process fails, it must be recorded to the log file specified by *-l* parameter as follows:

```
[time_stamp]: Registry file verification failed!
```

Then, the program must terminate. If the registry file verification is successful, the changes in the directory is checked. If there is no change in the directory, the following log is written to the log file:

```
[time_stamp]: The directory is checked and no change is detected!
```

If there are some changes in the directory, the changes are recorded to the log file as follows:

```
[time_stamp]: [Path_of_file] is [change_type]
```

In this log line, the *change_type* parameter can be one of the following values:

```
change_type = deleted | altered | created
```

2 Notes

1. You can ask questions about the experiment via Piazza group (piazza.com/hacettepe.edu.tr/fall2020/bbm465).
2. Late submission will not be accepted!
3. You must compile your program on Java version of the dev machine.
4. You are going to submit your experiment to online submission system:

   http://submit.cs.hacettepe.edu.tr/

5. The submission format is given below:

```
<Group id>.zip
-[src]/
      --*.[java]
```

3 Policy

All work on assignments must be done with your own group unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work (from internet), in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] Java keytool: https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html

[2] Package javax.crypto : https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html