Exercise 3 (Grad) -Authenticated transaction block (simplified Bitcoin)

Yuvaraj Sripathi

UFID: 14677313

Programming Language used – C++

1. Transaction Block

The program is built along with the previous assignment's code base – Ledger

Basic approach – All the ledger records are maintained in Linked list data structure and also makes use of maps to keep track of existing transactionIds, records.

Each of the node is the Linked list, contains the custom data structure for a record which has the following as its members,

- TransactionId
- Vector of utxos
- Vector of outputs
- Boolean flag to indicate the validity of the record
- Record count to check genesis etc.

To the Ledger interactive menu, 3 additional options/features are included to the previous Ledger assignment

- Output transaction block:

  From the ledger, consider all the transactions which are having valid signatures and also check to make sure that the transaction corresponding to the unspent transaction output is also having a valid signature or is already included in previous blocks. Couple of flags are included to the record data structure to indicate whether the given transaction record is already included in the block and signature verified or not. On selecting this option, all the valid transactions are output to the console window in the format => Errors if any, Series of <transaction , Signature> combination. In order to check validity of each of the transactions, the public key file of the owner needs to be present and this is maintained using a map data structure which contains keys as account names and values as valid filenames.

  If the verification is wrong or if for a transaction, public key of the utxo owner is not found, then that transaction remains in the ledger and any further transactions that refer to this transaction does not get into the current block.

Also, check if the valid transaction has only one owner account for all the utxo else display the corresponding error message.

- Read:

For reading the name of the account and name of the file containing the corresponding public key. On getting the input, check for file open and update the file and account name in a map – fileMap<string, string> which will be used for getting the latest file name for obtaining the public key during signature verification process.

- Check:

This takes the transaction ID as the input, if the transaction ID is found to be in the ledger and is a valid transaction, then the filename for the same is retrieved from the fileMap and check for signature verification and display “OK” or “Bad” accordingly. Also, check if the valid transaction has only one owner account for all the utxo else display the corresponding error message.

Crypto libraries used for implementation,

Openssl library is used.

In particular, EVP and RSA related APIs are used for verifying the signature.

Implementation files – cryptoUtil.h and cryptoUtil.cpp are added to the existing Ledger source code for crypto related operations.

NOTE: The signatures must be in base64 format (along with newline characters)

2. Wallet

The wallet program is built along with the previous assignment's code base – Ledger

Basic approach – All the ledger records are maintained in Linked list data structure and also makes use of maps to keep track of existing transactionIds, records.

Each of the node is the Linked list, contains the custom data structure for a record which has the following as its members,

- TransactionId
- Vector of utxos
- Vector of outputs
- Boolean flag to indicate the validity of the record
- Record count to check genesis etc.

To the interactive menu of the Ledger two additional options are included,

- Read

  For reading the name of the account and name of the file containing the corresponding private key. On getting the input, check for file open and update the file and account name in a map – fileMap<string, string> which will be used for getting the latest file name for obtaining the private key during signature verification process.

- Sign

  To obtain the private key file corresponding to the owner of the utxo and perform encrypt the SHA256 hash of the transaction record with the private key and display the same. The signature is stored in the record data structure for further use.

  The signing process failse, if there is no private key file for the fiven account name of the owner or if the transaction is not a valid one or if there are more than one utxo owner in the same transaction.

Crypto Implementation:

Openssl – EVP and RSA libraries are used for signing transaction record's hash and getting the signature.

Implementation files – cryptoUtil.h and cryptoUtil.cpp are added to the existing Ledger source code for crypto related operations.

NOTE: The signature is converted to base64 format to display and to store.

Issues/Roadblocks faced:

There was only one i.e. handling the hex format of the keys. I was not able to,

- Verify the transactions with the public key and the given hex format signature. The API "EVP_DigestVerifyFinal(m_RSAVerifyCtx, MsgHash, MsgHashLen)" returns false.
- Not able to display the output of the signature created using the private key and transaction record – Couldn't do it accurately via the program using openssl APIs

The main reason was that I couldn't find any useful information related to the above issue in the Openssl documentation.

FIX:

After discussing with TA's announcement, I performed all operations in Base64 format and all the options/functionalities work fine as per the given specification document.

Request you to please consider during grading this slight deviation from the norm.

Reference,

Most of the crypto related operations, I referred to the links given by TA and followed the same logic with Base64 format signatures.

Testing: (On storm server)

On using Base64 format signature in both txblk and wallet programs, all the features and functionalities work as per the specification document.

Have tested the following cases for txblk

- Read proper key files
- Read files not present in the system – displays error accordingly
- Check for signature verification – Returns "OK" if successful
- Check for signature with different public keys – Returns "BAD"
- Output block – Only the signature verified transactions enter the current block
- Output block – Non verified signatures are still present in the Ledger
- Output block – Every transaction's utxo transaction also need to be verified (or included in previous block) else the current transaction does not make it to the block
- Output & Check block – All the account names of the utxos in a given transaction needs to be of the same owner else report error and not include in the current block

Ex,

Input file (**NOTE – no signature for second record**) Actual file in txblk folder => test5.txt

```
9b27ea15; 0; ; 1; (Alice, 5000)
aa9j4DFjkGMrhg9jE6otLbejg0F/xpkd5Vi1lOqeSATHjb7XUOc4T4Kyk8e78Sto\nksXlRqkOpH6zpV5TCGH
qh3jqDuRs6RQxDFvAXMNcxOrSO/5X40MEcAfYq+BHNOec\n5XGHrl0/wzXG4dICO4x5K1uKV5KGltSlCM
OX+gBa5joAB5DH7K6/Hu/iY8myxhRw\nICsVykZ8aQFO+aIBH4U2CE9I4bKpTQ/w0jMvAAtF8SL16m3wx
RIYWo/fhpgy362e\nxjcmyOJB5kOqQf/DcvACOKlalVMIFHJEONKs+f1wg93dU1Bg5kwGVdDQCD5NVBN
b\nMP2W1ix0xOXCXizlO6NXGg==\n
cbdf7b37; 1; (9b27ea15, 0); 3; (Bob, 150)(Alice, 4845)(Gopesh,5)

cbe628f2; 1; (cbdf7b37, 0); 3; (Gopesh, 100)(Bob, 45)(Bob, 5)
HRibfCp/jWo76cb8zJD8hP79AL87r4ZkN9/6xykYPTZFwxoGsmbb0qFnOvxFcloA\nULz56rmXSJaCGoIov5
xsKHhqk1uRS20db7OicQAsixWWCc1E5Pd93Z2GdEBx6dbd\n5lyD7zPiU3sjDcJA1/sS1Mb/ksG2ANi8273
m1FhzFtzwlcXZzDtnn8MPhH6w1CeF\ntPDb0YZ4ycAsWB1tSu8zB+RXxuMxXbUWWrD5J1eFhFxBwdW+
0+/feghLP5KrCQmD\nmQ6XqhYET0+W4FVbJPvM+iWCR0usFnYwSCvmJa0H+ykJ5KJhygWRBVIz2ZhaeD
6m\n5NHtXbZLGzxu7gx8NKLo+Q==\n
```

```
storm:43% ./txblk
f
test4.txt
TransId: 9b27ea15, result - good
TransId: cbdf7b37, result - good
TransId: cbe628f2, result - good
r
Alice alicepublic_key.pem
Read <account name> and <keyfilename> sucess
r
Bob bobpublic_key.pem
Read <account name> and <keyfilename> sucess
c
9b27ea15
OK
ccbdf7b37
Invalid option. Try again
c
cbdf7b37
Bad
o
1
0; ; 1; (Alice, 5000)
aa9j4DFjkGMrhg9jE6otLbejg0F/xpkd5Vi1lOqeSATHjb7XUOc4T4Kyk8e78Sto
ksXlRqkOpH6zpV5TCGHqh3jqDuRs6RQxDFvAXMNcxOrSO/5X40MEcAfYq+BHNOec
5XGHrl0/wzXG4dICO4x5K1uKV5KGltSlCMOX+gBa5joAB5DH7K6/Hu/iY8myxhRw
ICsVykZ8aQFO+aIBH4U2CE9I4bKpTQ/w0jMvAAtF8SL16m3wxRIYWo/fhpgy362e
xjcmyOJB5kOqQf/DcvACOKlalVMIFHJEONKs+f1wg93dU1Bg5kwGVdDQCD5NVBNb
MP2W1ix0xOXCXizlO6NXGg==

p
cbdf7b37; 1; (9b27ea15, 0); 3; (Bob, 150)(Alice, 4845)(Gopesh, 5)
cbe628f2; 1; (cbdf7b37, 0); 3; (Gopesh, 100)(Bob, 45)(Bob, 5)
```

Have tested the following cases for wallet

- Read proper key files
- Read files not present in the system – displays error accordingly
- Try Signing without the owner's key for a transaction – reports error accordingly
- Sign the transaction with the utxo owner's private key – Display the signature and store it the record corresponding to that transaction
- Sign request for a transaction not in ledger – Report error accordingly

Ex,

Input file – (Actual file in wallet folder => test4.txt)

```
9b27ea15; 0; ; 1; (Alice, 5000)

cbdf7b37; 1; (9b27ea15, 0); 3; (Bob, 150)(Alice, 4845)(Gopesh,5)

cbe628f2; 1; (cbdf7b37, 0); 3; (Gopesh, 100)(Bob, 45)(Bob, 5)

cbe628d2; 2; (cbdf7b37, 1)(cbdf7b37, 2); 1; (Bob, 4850)
```

```
TransId: 9b27ea15, result - good
TransId: cbdf7b37, result - good
TransId: cbe628f2, result - good
TransId: b124884b, result - good
r
Alice aliceprivate_key.pem
File - aliceprivate_key.pem read success
Private key <account name> , <keyfilename> read
r
Bob bobprivate_key.pem
File - bobprivate_key.pem read success
Private key <account name> , <keyfilename> read
s
sad67sd
TransId: sad67sd not found in the ledger
s
9b27ea15
Signature(Base64 format): aa9j4DFjkGMrhg9jE6otLbejg0F/xpkd5Vi1lOqeSATHjb7XUOc4T4Kyk8e78Sto
ksXlRqkOpH6zpV5TCGHqh3jqDuRs6RQxDFvAXMNcxOrSO/5X40MEcAfYq+BHNOec
5XGHrl0/wzXG4dICO4x5K1uKV5KGltSlCMOX+gBa5joAB5DH7K6/Hu/iY8myxhRw
ICsVykZ8aQFO+aIBH4U2CE9I4bKpTQ/w0jMvAAtF8SL16m3wxRIYWo/fhpgy362e
xjcmyOJB5kOqQf/DcvACOKlalVMIFHJEONKs+f1wg93dU1Bg5kwGVdDQCD5NVBNb
MP2W1ix0xOXCXizlO6NXGg==

Signing transaction success
s
cbdf7b37
Signature(Base64 format): rOye1W+ABzoce9IVMrAGKqVAP+diFQwbSY6UUyE0aolx9QM8juSWf/IO3njzOjbZ
7jULIIo9uUvIS0bFJEwSSLCI7JYfqL51eaOUZbInEpU+WS0A1oFDkz97BJC4SLa4
f7XFm6V4BxVv/Jm8VLd01AgKGBTcqbXkXuSjz2ANZN7x4GXrVDkhIah+KwH1LaXh
GAj95NyoqA574OSm5cnhVh3QUOwVBWBqp/rQOm6FmPiKtvepG4GF2HIAf3Fo1+aV
2fXi87GID8RBMpqKi7phsevIEnOpOxiDfcSx4azdbWitM8d1U8PPZB4BUnbPsX2I
kyHaqhmRCSbYJCSeDYLaGg==

Signing transaction success
s
cbe628f2
Signature(Base64 format): HRibfCp/jWo76cb8zJD8hP79AL87r4ZkN9/6xykYPTZFwxoGsmbb0qFnOvxFcloA
ULz56rmXSJaCGoIov5xsKHhqk1uRS20db7OicQAsixWWCc1E5Pd93Z2GdEBx6dbd
5lyD7zPiU3sjDcJA1/sS1Mb/ksG2ANi8273m1FhzFtzwlcXZzDtnn8MPhH6w1CeF
tPDb0YZ4ycAsWB1tSu8zB+RXxuMxXbUWWrD5J1eFhFxBwdW+0+/feghLP5KrCQmD
mQ6XqhYET0+W4FVbJPvM+iWCR0usFnYwSCvmJa0H+ykJ5KJhygWRBVIz2ZhaeD6m
5NHtXbZLGzxu7gx8NKLo+Q==

Signing transaction success
s
b124884b
TransId: b124884b has multiple owners for utxo
Sigining transaction fail
```