Design.doc

The system stores keys and pins inside of ssBank.bin and ssATM.bin. The account balances are also stored in these super secret files. We implement our encryption schemes right before and after we send all data. we print the same messages as project 1 except for in router.py. In router.py we made it so that it tells if a message is sent from the bank or atm. The cards contain only a name. Probably not the safest but it's honest work for honest people.

First the user must start a session each of Router.py, Bank.py, and atm.py.
The user will then copy the contents of a "user.card" into a file "inserted.card".
The User will go into the atm.py session and enter the command "begin-session".
begin-session will read the contents on the card and prompt for the user
to confirm his/her identity with a four digit pin.

The ATM will then grant access to the User's bank account. The User may now enter one of three commands. The withdraw command is the only command with a supporting argument. The "amt" will be the monetary amount(in digits) that the user wishes to withdraw.
balance
withdraw amt
end-session

typing "end-session" into the atm prompt will and the user's session, therefore causing them to "begin-session" again before any more actions may be taken.

Our ATM machine adheres to the fail-safe default principle because no withdraws will be attainable if there is no connection with the bank machine. The bank machine holds the balance information.

Our ATM machine also carries a least privilege mechanism. The user at the atm will not be able to even access the commands the atm has to offer unless they are signed into their account. The whole system changes accounts along with the user, so they may not effect others
accounts.