1. For security, I decided to go with a symmetric encryption model for two reasons - one, it's relatively easy to implement, and two, without access to the .atm and .bank files, it is relatively safe to store the same password in both files. Anyways, so I encrypt each message using an XOR encryption method using the keys in the .atm and .bank files to encrypt it, and then decrypt it on the other side, and vice versa. I also attempted to hash the pin using a salt and stored it in the .card file. I also have extension error handling and checks in place for invalid input.

2.

1) Brute force attack - first, the key generated is 48 characters long and although brute forcing the key is possible, it's not probable to guess it. As for stuff like the pin, brute force attacking it is possible, and I guess one way to defend against that would be to block attacks after a certain number of attempts - I just haven't had time to implement that.

2) Invalid input - this attack is something that I tried to address through extensive error checking.

3) Buffer overflow - I have tried to structure my data such that buffer overflow attacks would be very hard to accomplish.

4) MIM - This attack, given the router, is the one I was the most concerned about. However, with no access to the secret keys, this attack is nigh impossible to accomplish, without brute forcing it which would take an extremely long time to do since each key is 48 characters and randomly generated.

5) Card attacks - I didn't actually end up using the card for much security. I just kind of checked for it's existence and left the actual handling of data.

3. There were definitely more attacks, I just didn't have time.