Maddie Zug
System Security
Fall 2018
A2

Rationale

Gen
I used pycryptodome's Crypto library to generate the RSA public and private keys for Bob and Alice. In Alice's file I used the same library's get_random_bytes method to generate a cryptographically secure session key.

Handshake
To implement the key exchange protocol I used RSA 2048 with OAEP padding to encrypt the required parts of the message specified in the instructions:

```
1. A -> B: B, tA, Enc(A,kAB; K_B), Sign(B, tA, Enc(A,kAB; K_B); k_A)
```

Bob uses the Crypto library to decrypt the session key and necessary information with his private key.

Unsafe Mode
No encryption was used. Messages are sent with a message number.

Enc Mode
For encrypting messages I used Crypto.Cipher's AES library. This gave me access to methods that would pad my messages and generate IVs to use to send with messages. I used the shared session key sent with the key transfer protocol to do this encryption.

Mac Mode
For signing messages I used Crypto.Hash's CMAC mode which uses a block cipher to generate a MAC. I then send that MAC as a tag attached to the unencrypted message.

Enc and Mac Mode
This mode uses the same methods from Enc and Mac, encrypting the message first using AES and then using the Hash library to sign the whole message.

Justification
I chose to use AES 256 in CBC mode for encryption because it is widely considered secure and hard to break. To encrypt the session key I used RSA 2048, again because it is widely accepted as secure. I used CMAC signing because it works with AES and had good documentation.

Specification

I chose to have Mallory automatically forward the handshake message to Bob, assuming that if our protocol worked she would not be able to initiate any attacks on that message. I also chose to send message numbers unencrypted as part of a tuple, which means that Mallory can be deceptive to Bob if a message is modified or replayed. For replay attacks I chose to have Mallory save all past messages in a dictionary with their message number so she can specify which one to send along.

External libraries

I used pycryptodome's Crypto library to implement RSA, AES, MACs and hashing.

Known problems

When bob receives the handshake method he checks the timestamp but does not currently check the signing of the handshake method. This opens him up to attack. If I had more time I would fix the bugs in the signing code. I would also go back through the code and make it more robust. I did some error handling but there are definitely inputs that would crash the program that would be helpful to fix. If you send multiple messages from Alice to Mallory some of the messages will get lost. It is unrealistic to expect that Alice would patiently wait for Mallory to make a decision about a message before sending another one, so if I had more time I would want to modify my socket code so that Alice could send multiple messages in a row without waiting for Mallory's actions. Finally, Bob could do a better job of verifying message numbers to resist replay attacks.