**CS 4732/5732 Homework #7**

***Due electronically by midnight August 4th, 2025***

For submission, if done on paper please scan and submit as a pdf. If done in word, please submit the .docx or .doc format.

**IMPORTANT**: Clearly indicate outside resources utilized and sign below. Failure to cite use of outside resources will be reported for appropriate disciplinary actions. Note that discussions with other students are encouraged; copying – with or without modifications – is unacceptable and will also be reported.

I discussed one or more problems with the following people:

I hereby certify that any outside resources utilized, other than the textbook and class materials, are clearly cited. All other material I provide for this homework submission is my own original work.

Jamie Harris

*Printed name*

1. (6 points) Session keys and master keys are private and usually no more secure than each other. So why would it be fine to use a master key to distribute session keys? How do we gain anything?

With a master key leading to a key distribution center, an individual no longer needs to store keys for each person they want to communicate with. Instead they only need to store one key, the master key. Additionally, because you typically want to routinely change your key in a symmetric encryption scheme, having the KDC generate a new key for each session is much more efficient than having to personally generate a new key and send to the other person.

1. [12 points] a) What is the difference between a digital signature and a MAC?

MACs are used to make sure a message has not been modified by a third party during delivery.

Digital signatures are used when you want to make sure that a message is not modified by the sender/recipient after the fact.

1. Give an example of something you would use a MAC for but not a digital signature.

If you are sending out a press release as a PR representative, and you want to make sure that the message is not altered by a rival company.

1. Give an example of something you would use a digital signature for but not a MAC.

If you are trying to buy product rights from another company, and want to make sure they do not try to change the sale price after the deal has been made.

3. (4 points) What is stored in a bitcoin wallet?

Public/private key pairs

1. (8 points) If in bitcoin, addresses are derived from public keys, isn’t that a weakness? Could someone use one of your public keys to generate a bitcoin address? If so, could they do anything nefarious with that? Explain your answer on what you think the limit of what they could do with it.

Even if someone was able to get your public key and get a new address, they wouldn’t really be able to do much with it because the wallet tied to that public key would still need its private key to access it, so the address would not do them any good if they don’t have your private key.

5. (6 points) Compare and contrast Public Key authorities with a Public Key directory.

Public key directories are basically just a global look up for a person’s public key, which anyone can access. Public key directories fall short when it comes to security, as there is not inherently a way to make sure that someone’s key is not being modified by another party. Public key authorities are similar to key distribution centers. Party A will send a request for Party B’s public key to the authority, which will then send a message with Party B’s public key and Party A’s original request back to Party A. This message will be encrypted with the authority’s private key, which allows Party A to be sure that the B’s public key is legitimate.

6) (6 points) In the key exchange protocols that we saw, many of them used nonces. Yet these nonces are not shared and agreed upon, instead one of the parties picks one and then sends it to the other. What security in particular are these nonces used for?

Nonces are used to make sure that the session key was not intercepted during transmission, and to authenticate that the sender and receiver are intended targets.

7) (6 points) Using RSA for a digital signature, the message is hashed and then the hash is encrypted using RSA to create the signature. Why not just encrypt the entire message instead? It sure seems to be a more straightforward way to sign the message.

It allows for better confidentiality, because then you are able to just send the plaintext message and its signature to a third party to verify if a dispute occurs. If you encrypt the message before signing, you would have to also provide the third party with the decryption key