Comp 5370

Answers for Test 1

1. Confidentiality, Integrity, Availability, Non-Repudiation.
2. Confidentiality is importing because there is important information such as a users address or credit card info being inserted into the system which would cause a problem if that information was made accessible to the public (ie. Stolen credit cards, strangers getting list of home addresses).

Integrity is important because the user expects accurate information in order to have trust in the systems data and reliability.

Availability is important for the user because this will erase limits of geographical and real time accessibility which will increase consumer traffic opportunities.

Non-repudiation is important because both parties are agreeing to terms of sale which neither user nor server can refute.

1. Confidentiality should be protected against someone like Eve, someone who just wants to passively look or listen to the messages being sent, because this person would be able to read the messages/text and then leave the channel and do something else with that secret information.

Integrity should be protected against someone who is looking for this information passively or maliciously (essentially everyone) since they may try to alter the text before it is then sent through to the original receiver.

Availability should be protected against a malicious attacker who is looking to perform a possible DoS attack or something that could possibly bring down the server, this person could perhaps break down the server and we would then lose traffic time (this could be helped with Resiliency, but I chose not to implement due to cost constraints).

Non-repudiation should be protected against any user (or someone disguised as a user) who is trying to get out of a previous purchase or even trying to get in the middle of another user’s previous purchase for their own intent.

1. Anonymity, Resiliency, Forward-Secrecy.
2. Without Anonymity the personal details of any and all users are left unsecured and unprotected against any malicious attacks.

If Resiliency is compromised a potential attack against the server could lead to longer down times of the server, thus resulting in server traffic to stop.

Without implementing Forward-Secrecy, any future attack could result in the loss of data and information recorded from server history.

1. Anonymity is not worth protecting because the company must be able to corelate between the user and their account information/personal history.

While Resiliency would be a convenient implementation, it is not crucial to the overall functionality of the server.

Forward-Secrecy is not worth protecting because with the implementation of confidentiality, the secrecy of all users at all times should remain true, which makes the implementation of Forward-Secrecy unnecessary.

1. This is not a safe protocol because:

AES256\_ECB is wrong because ECB is too easily translatable by merely observing the contents of the cipher text.

This impacts the confidentiality of the message.

Change the AES256 from ECB to Counter mode (CTR).

HIGH

This is a safe protocol because:

Confidentiality comes from the AES256\_GCM\_SIV since the GCM is an upgraded version of CTR for AES256.

Integrity comes from the AES256\_GCM\_SIV, due to its built in integrity checking, and also comes from the Sign RSA, since the RSA functionality results from the modification of a public key (whoever the other is in this instance) and a private key (your key), and the HMAC-SHA3 function which simply encrypts the text.

Authentication comes from the RSA sign since the sign uses a system of private and public keys to encrypt/decrypt the message.

This is a safe protocol because:

Confidentiality comes from the AES256\_CTR 🡪 HMAC\_SHA256 because the AES256\_CTR brings Confidentiality/Integrity, and the HMAC\_SHA256 brings simple Integrity.

Integrity comes from the AES256\_CTR 🡪 HMAC\_SHA256 because the AES256\_CTR brings Confidentiality/Integrity, and the HMAC\_SHA256 brings simple Integrity, and Integrity also comes from the ECC Sign.

Authentication comes from the ECC sign since the sign uses a system of private and public keys (similar to the RSA sign) to encrypt/decrypt the message.

This is a safe protocol because:

Confidentiality comes from the AES256\_GCM because the GCM is an upgraded version of CTR for AES256, which is Confidential since we use the CSPRNG for the key and PRNG for basic nonce.

Integrity comes from the AES256\_GCM because the GCM has built in integrity checking and from the ECC Sign using a system of public and private keys.

Authentication comes from the ECC sign since the sign uses a system of private and public keys (similar to the RSA sign) to encrypt/decrypt the message.

1. Scenario 1 is not a proper two-factor authentication because both things needed to prove authentication are things the user knows. While both of these are independent passwords, it could be possible that if someone was able to get one password, they would probably be able to get the other.

Scenario 2 is a proper two-factor authentication because the HOTP device is a MFA One time code device (something you have), and the fingerprint is a biometric password (something you are).

Scenario 3 is a proper two-factor authentication because a code from hardware device is essentially a security token (something you have), and the PIN is a form of password (something you know).

Scenario 4 is a proper two-factor authentication because your voice itself is a biometric password (something you are), and what you say is a password (something you know).

1. They must first have their shared P and G numbers, then they will both get their own secret number (a and b; this settles what is needed for the Diffie-Helman). Then we must also get the public and private keys used for the RSA sign (used for authentication).
2. Don’t roll your own crypto!
3. I originally assumed it had to do with how you open and edit files for grading purposes. Maybe even because this is a software class and we only work with file editors that are simple. But now I guess it may have to do with different ways PDF’s Doc’s or whatever other text editors encrypt their data, and maybe the very simple text editors do not encrypt their data much, if at all.
4. No (sketchy)
5. No
6. The questioning brought up by real cryptographers makes me skeptical myself, and after doing a bit of research after and seeing other cryptographers question if there is a ‘backdoor’ or not makes me too skeptical to use it.
7. If integer factorization becomes easy, then the heart of RSA signing is no longer valid. This is due to RSA signing depending on the large amount of computing time it would take to compute the integer factorization for such a large number. In this protocol the loss of RSA would make the protocol lose authentication. This means the protocol is now subject to attacks from Mallory since the messages do not have any authentication.
8. If HMAC is broken not needing the key, then we would not need the key to unhash the MAC. This means the raw message could be seen by simply coming up with a new key and using it on the HMAC. This is a big problem for the protocol it would now lack Integrity, Confidentiality, and Authenticity, since now the message could have been decrypted by others, or a new message could be generated, encrypted, and sent.