Assignment 5
Gunnar Yonker

1.

(1)

Personal computer:

6 characters: (1000 hash calculations * 128^6) / (10^9) = 4,398,046 seconds or about 50.9 days

7 characters: (1000 hash calculations * 128^7) / (10^9) = 562,949,953 seconds or about 17.85 years

8 characters: (1000 hash calculations * 128^8) / (10^9) = 72,057,594,037 seconds or about 2285 years

9 characters: (1000 hash calculations * 128^9) / (10^9) = 9.22^12 seconds or about 292,471 years

Supercomputer:

6 characters: (1000 hash calculations * 128^6) / (10^15) = 4.4 seconds

7 characters: (1000 hash calculations * 128^7) / (10^15) = 562.9 seconds

8 characters: (1000 hash calculations * 128^8) / (10^15) = 72,057.6 seconds or about 20 hours

9 characters: (1000 hash calculations * 128^9) / (10^15) = 9,223,372 seconds or about 106.75 days

(2)

6-9 characters, potential number of passwords for each employee:

So, for each number of characters on a supercomputer with each hash operation taking 1000 calculations and then checking to see if it matches any of the hashed passwords in the password file.

1000 hash calculations * (128^6 + 128^7 + 128^8 + 128^9) / (10^15) = 9,295,997 seconds or about 107.6 days

It would take about 107 days for the supercomputer to find out all 100 passwords if the supercomputer could do 10^15 calculations per second and the hash operation takes 1000 calculations.

(3)

There is 1 million passwords in the dictionary and each would take 1000 calculations for the hash operation.

1000 calculations per hash * 1 million dictionary words = 1 billion calculations

1 billion calculations / 10^9 (regular computer) calculations = 1 second

Using a dictionary attack the hacker would be able to get around ¼ most likely or 25% of the passwords among the 100. According to this probability, the hacker will get around 25 of the 100 passwords using this attack.

2.

(1) Now the salt needs to be added for each individual password hashing which will add additional time because each password needs to be checked with its specific salt.

Assignment 5
Gunnar Yonker

100 passwords * (128^6 + 128^7 + 128^8 + 128^9) * (1000 calculations per password) / (10^15 calculations per second) = 929599697.889 seconds = about 29.5 years

I don't think an efficient search method will make a difference in this case because if the brute force attack is going to take 29.5 years in this case, even with an optimized search method this is still way too long of a time investment.

(2)

(1000000 * 100 * 1000)/(10^9) = 10,000 seconds = ~2.78 hours

Using the dictionary attack of 1 million words, 1000 calculations per hash, and for each of the 100 passwords using the personal computer(10^9 calculations per second) would take 10,000 seconds which is about 2.78 hours. The hacker could expect to find around 25% success again, so 25 of the 100 passwords.

3.

a.

A Message Authentication Code(MAC) is a cryptographic tag that is used to verify the authenticity and integrity of a message during communication to ensure the information is being exchanged with the authentic user not an attacker. It is generated using a secret key and a hash function to create a fixed-length message. The sender would hash the message using a hashing function such as SHA-3 then the sender would apply a key to encrypt the message. The receiver of the encrypted message would then hash using the same hash function and key, the resulting MAC would then be compared to the one that was sent with the message from the sender. If they match, then the message is considered to be authentic.

b.

Digital signatures have the advantage over message authentication codes because they provide non-repudiation which means that the signer is unable to deny that they signed the message. The digital signature is signed using the users private key and the receiver uses the public key to verify the signature. The receiver is then able to verify that the message is authentic and that the sender is able to be identified as well. MAC is only able to verify that the message is authentic, not the identity of the sender whereas digital signatures are able to do so.

c.

Post-quantum cryptography is relevant for any cryptographic algorithm that is using public key cryptography and key exchange protocols. This includes RSA, Diffie-Hellman, and digital signatures like ECDSA and RSA. All of these algorithms are relevant to post-quantum cryptography because it can allow these algorithms to become more complex and more difficult to break.

d.

Lattice-based cryptography: This approach is based on hard mathematical problems related to lattice theory.

Assignment 5
Gunnar Yonker

Code-based cryptography: This approach is based on hard mathematical problems related to error-correcting codes that allows data to be transmitted over noisy channels.

Multivariate cryptography: This approach is based on hard mathematical problems related to solving systems of multivariate polynomial equations.

Hash-based cryptography: This approach is based on using cryptographic hash functions to generate digital signatures and key exchange protocols that can become resistant to quantum attacks.