Vincent Yasi

CS 370 -Intro to Security-

Problem Set 3

Cryptographic Hashes & Message Authentication Codes (MACs)

Q1[3 pts]: What are the three key properties of a cryptographic hash?

That it is Efficient, meaning it is easy to compute the hash, given the inputs.

That it is Pre-Image Resistant, meaning given the hash, it is infeasible to compute the original input that gave the hash.

That it has Collision Resistance, meaning if given the input to the hash, it is infeasible to compute a second input that will give the same hash (weak collision), as well as it is overall infeasible to find two inputs at all with the same hash (strong collision).

Q2 [3pts]: What is a birthday attack? Consider a hash function that maps inputs to a 32-bit hash. If an attacker launches a birthday attack, approximately how many steps will it take the attacker to find a collision with a 50% probability of success?

A Birthday Attack uses the Birthday Paradox to try and find collisions in the hash function. The Birthday Paradox states that there is a certain probability that two values chosen from the same set will be equal (there will be a collision) based on how many samples are chosen from the set. As the sample size increases, so does the probability.

With the Birthday Paradox, it states that there is a 50% chance of collision with a sample size of approximately half the total size of the set (m/2). For a 32-bit hash (2^{32}), this would be approximately $2^{(\frac{32}{2})} \rightarrow 2^{16}$. Therefore, it would take about 2^{16} steps to get a collision.

Q3 [4 pts]: What is the difference between a cryptographic checksum and a message authentication code? What primitive should one use to integrity protect files being transferred on an open channel?

A checksum (hash) shows if the message it is tied is valid for the checksum, as the checksum is calculated from the message. However, it does not use a key, and so can usually be easily recalculated if the message was changed, being easily defeated. By itself, all a checksum does is tell if the checksum sent is the one for the message sent.

A MAC, meanwhile, uses a key and is essentially an encrypted hash. Because it is encrypted, it cannot be easily modified by someone who does not know the key, and so better protects against modification of the message. A MAC is better at protecting the integrity of the message, as presumably only someone who knows the key for the message would eb able to modify the MAC.

Public-Key Cryptography (Diffie-Hellman, RSA, Digital Signatures)

Q4 [3 pts]: Name three differences between secret-key cryptographic schemes and public-key cryptographic schemes?

With secret-key schemes, the same key is often used for encryption and decryption, while a public-key scheme has a public key used on one end and a private one on the other end.

Because of this, another difference between the two is that both parties in secret-key (often) need to exchange the key in a secure way beforehand, but with public-key, one party can simply look up the public key of the person they are sending to and encrypt with that, bypassing needing to exchange keys in a way that keeps them secret and is done beforehand.

A last large difference is in overhead cost and time needed for the two schemes. With secret-key, it is relatively simple to encrypt/decrypt messages, as the algorithms and keys are simpler (the "strength" is contained in the fact the key is secret). However, with public-key, the keys and algorithms are more complex to account for the properties of the system.

Q5 [3 pts]: What is a digital signature? What security properties does it provide?

A digital signature is a way to mark, or otherwise indicate, the origin of a message, done in such as way as it cannot be easily forged or computed, but can be easily verified on the receiving end. It primarily provides authenticity for the message, as it verifies who sent the message.

Q6 [3pts]: How are digital signatures different from MACs? Contrast the security properties they provide.

A digital signature provides verification of who sent the message, while a MAC provides proof that a message was not changed in transit and is as it was sent. Digital signatures provide authenticity to a message and its originator, while a MAC provides integrity to the message and that it is as it was sent. The first can be thought to verify the source of a message, while the second verifies the content of the message. A signature doesn't tell you anything about the message itself or if it is valid, only who sent it. Meanwhile, a MAC doesn't tell anything about who sent the message, only if it is the same message that produced the MAC in the first place.

Q7 [9pts]: Alice owns a public-private key pair (PKA, SKA); Bob owns a public-private key pair (PKB, SKB); Assume that they know each other's public keys and answer the following questions:

If Alice wants to send a secret message M to Bob, what should she do? Show what needs to be transmitted using the notation used in class.

If Alice wants to send a message to Bob, she should encrypt the message using Bob's public key. Then, only he can decrypt it, as only he knows his private key that is paired to the public key.

To do this, for message M, Alice would use: $F(M, PKB) \rightarrow M^{PKB} \ mod \ n$

Bob receives a 128-bit AES key and the message "from Alice: use this key to send me your credit card number", both enciphered with his public key. Should Bob do what the message says? Assume Bob does want to send Alice his credit card number. If yes, why? If not, how should the message have been enciphered?

Bob should not do what the message says, as it might not actually be from Alice. It could be a man-in-the-middle attack from Eve. As the key (which is just a key that anyone could have generated) and the message are encrypted with Bob's public key, anyone could have done it (such as Eve). There is no proof it was Alice who sent them (there is no protection of Authenticity).

To verify the message and key is from Alice, they should both be encrypted with Alice's private key. Bob can then decrypt them with Alice's public key. If they decrypt with Alice's public key, that proves they were encrypted with Alice's private key (which presumably only Alice has access to). This provides Authenticity.

If M is a really long message, how should Alice transmit the message while keeping it secret and minimizing the effort? Please explain.

If M is long, encrypting the whole message with a public key scheme will often be costly, as they are much less efficient than private key schemes. To minimize effort, the message itself should be encrypted with a cheaper private key scheme. Then, the key for the private key scheme should be encrypted and sent using the public key scheme. The key will be smaller than the whole message, and so easier to encrypt in the public key system. Once encrypted, it can be securely sent, and the private key encrypted message can be sent and decrypted with said key.

By using the costly public key scheme on the small private key scheme key, and the cheaper scheme on the lengthy message, it reduces the overall cost to send the message securely.

Q8 [3 pts]: Do digital signatures and MACs increase the length of message to be transmitted? Explain Why?

They do (generally) increase the length of the message, as they often are concatenated onto the end or beginning of the message itself. There are certain ways to avoid this, such as using the last block of the message itself as the MAC or otherwise integrating the MAC/signature into the ciphertext without increasing the length. Most methods, and the simpler ones, do increase the overall length of the message.

Q9 [3 pts]: Using the notation from the class, show how a message m is signed with an RSA key-pair (N, d, e).

A message can be signed with an RSA key-pair by concatenating a hash encoded with the private key to the end of the message. This hash can be decoded using the public key, and the hash validated. If it is valid, then it was signed with the user's private key (which presumably only they know).

This is done with: $m||\{h(m)\}\_(N, d)$, where (N,d) is the user's private key. (e, along with N, is the public key that can used to validate the signature by decoding the hash).

Q10 [4pts]: Contrast man-in-the-middle and meet-in-the-middle attacks.

A man-in-the-middle attack has a person intercept incoming messages from one end, do what they want to do with the messages, and then send out modified messages back out. The attacker poses as the intended recipient and/or the sender, and by doing so can modify the data stream.

A meet-in-the-middle attack, meanwhile, entails calculating an encryption scheme from both ends, looking to meet at some point in the "middle" to compare and contrast values from either side of the computation to determine the correct values, etc. of the algorithm, and overall break the encryption.

The first attack involves intercepting data and deceiving the sides in a transaction. It learns information by using this deception. The second attack, however, involves brute forcing the information they have (such as ciphertext and possible plaintexts) to be able to draw conclusions and information in the middle of the two operations. Man-in-the-middle attacks the authenticity, integrity, and accountability of data, while meet-in-the-middle attacks the security of the algorithms used.

Q11 [3pts]: Is it important to hash the message for digital signatures?

When signing a message, it is often important to hash it, as in order to verify a signature, you should not have to decrypt the whole message (and you don't want some one to). If you hash a message, and then sign the hash, a person looking to verify the signature only needs to use your public key and calculate the hash and validate it is correct, which are easy to do, do not require decrypting the message, and do not reveal or modify the content of the message itself.

As well, due to the way signatures work (primarily the public/private key relation), if the hash is not signed, and rather the message itself were signed, then the message could be decrypted without needing to know the private key at all.

Q12 [3 pts]: Does the hash function used in an RSA signature need to be a keyed hash function? Why or why not?

The hash used does need to be a keyed hash, as your private key needs to be used to encrypt the hash in order to sign it. This is what allows someone to use your public key and verify the signature. If the hash were not keyed, there would be no way to sign it, and anyone could include a valid hash (as that is one of the properties of a hash) which could be decoded by the message receiver.