Practice Questions for Computer Security

CS370/ECE499

Hash Functions and MACs

1. What are the three key properties of a cyrptohraphic hash?
   [Bonus] Which of the three properties implies the others. Please explain.

   Pre-image Resistance
   Weak-Collision Resistance also called 2nd Pre-image resistance
   Strong Collision Resistance

   In weak-collision resistance an adversary is asked to find a hash collision for a given input value. That is given x find x' such that h(x) = h(x') and x =/=x'

   In strong-collision resistance an adversary is just asked to find a hash collision. He can pick both x and x' such that h(x) = h(x') and x =/=x'.

   Strong ➔ Weak: Strong-collosion resistance property implies weak-colision resistsance. This can be shown as follows:

   Let a hash function h be strong-collision resistant. That is, no polynomial adversary can find x and x' such that h(x) = h(x') and x =/=x' with more than negligible probability. Now let us assume that this function does not have weak-collision resistance. That is, there exists a polynomial adversary who when given x can find x' such that h(x) = h(x') and x =/=x' with non-negligible probability. Then we can use this adversary to break strong collision resistance as follows. When asked to find a pair x and x' such that h(x) = h(x') and x =/=x', we pick an x at random and pass it as input to the polynomial adversary who can break weak-collision resistance. With non-negligible probability that adversary will return x' such that h(x) = h(x') and x =/=x'. Thus we can break strong collision resistance of h which leads to a contradiction.

   Weak-Collision Resistance ➔ Pre-Image Resistance: Under certain conditions weak-collision resistance implies pre-image resistance. This can be shown as follows:

   Let a hash function h be weak-collision resistant. That is, given x no polynomial adversary can find x' such that h(x) = h(x') and x =/=x' with more than negligible probability. Now let us assume that this function does not have pre-image resistance. That is, there exists a polynomial adversary who when given y can find an x such that h(x) = y with non-negligible probability. Then we can use this adversary to break weak-collision resistance as follows. When given x and asked to find an x' such that h(x) = h(x') and x =/=x', we compute y = h(x) and pass it as input to the polynomial adversary who

can break pre-image resistance. With non-negligible probability that adversary will return x' such that h(x') = y. If x =/= x' then we broke weak-collision resistance of h which leads to a contradiction. But this only works when x =/= x'. When there are many inputs that map to the same output for a hash function then is quite likely that x =/= x'. Thus, if h is such that for every output, many inputs map to that output then it is highly likely we will get x =/=x'.

2. 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 is a type of cryptographic attack that exploits the mathematics behind the birthday problem in probability theory. The attack depends on the higher likelihood of collisions found between random attack attempts, as described in the birthday problem/paradox.
   Problem Solution:
   2^(n/2)
   n=32
   2^(16) steps

3. 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?

   Cryptographic checksum is keyless while MACs require a cryptographic secert key. Cryptographic checkssums are useful to guarantee the integrity of data when the checksum cannot be modified by the adversaty – for example when posted on a trusted and secure public website. Whereas, MACs are useful to guranatee the integrity of the data being transmitted on open channels as the MAC cannot be modified without the knowledge of the key. For this reason MACs also provide origin authentication as only the person holiding the key can create a correct MAC.

   The primitive that should be used is MAC.

Public-Key Cryptography and Digital Signatures

1. Name three differences between secret-key cryptographic schemes and public-key cryptographic schemes?

   Symmetric key cryptography uses the same key for both decryption and encryption and

both parties share the same key. Public-key cryptography uses different keys to encipher and decipher, and the sender and receiver do not share the same secret key. Symmetric key cryptography is much faster than public key cryptography, is easier to implement, and generally requires less processing power.

The security of public-key cryptographic schemes can be reduced to the difficulty of solving well-known hard problems.

2. Alice owns a public-private key pair (PK_A, SK_A); Bob owns a public-private key pair (PK_B, SK_B); Assume that they know each other's public keys and answer the following questions:

    a. 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.
    Alice needs to encipher the message with Bob's public-key (PKB, shown as PK_B below) so that only Bob who knows the corresponding secret or private-key SK_B can decipher it.

    Alice ---- $\{m\}_{PK\_B}$---→ Bob

    b. 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?
    No. Since the message was enciphered using Bob's public-key there is no guarantee that it was in fact Alice who sent it. Anybody could have used Bob's public key to generate that message. For, Bob to know that the message was from Alice, the generation of the message should involve Alice's private key that only she knows. An example is as follows:

    $\{K||MSG\}_{PK\_B}$, $\{h(\{K||MSG\}_{PK\_B})\}_{SK\_A}$

    c. If M is a really long message, how should Alice transmit the message while keeping it secret and minimizing the effort? Please explain.
    One example: $\{K\}_{P\_B}$, $\{M\}_K$, $\{h(M)\}_{SK\_A}$

    The bulk of the message is encrypted using a randomly generated symmetric key, and the symmetric is transmitted using the public-key. This is called hybrid encryption. The last block is for integrity protection.

3. How are digital signatures different from MACs?
MACs can be used to protect the integrity of the message (i.e. it has not been tempered with) and provide authenticity of origin (i.e., sent by X) to the receiver. While the

receiver can be sure that the message came from the claimed sender as only he and the sender share the secret key, the receiver cannot prove to a third party that the sender sent the message as the receiver has the capability to generate the message as he knows the secret key. Digital signatures can also be used to protect the integrity of the message (i.e. it has not been tempered with) and provide authenticity of origin (i.e., sent by X) to the receiver. However, since a digital signature is generated using a key only the signer knows, a receiver can prove to a third party that the sender sent the message. Thus digital signatures provide non-repudication – that is a sender cannot repudiate having generated the message.

4. Contrast man-in-the-middle and meet-in-the-middle attacks.
   Meet-in-the-middle attack: refers to a brute force attack that attacks the problem from both ends to trade computaional effort with space.

   Man-in-the-middle: referes to an active attack where an attacker sits in the middle of a conversation of tow legitimate parties and prentends to be the receiver to the sender and vice-versa.

5. Is it important to hash the message for digital signatures?
   RSA Signature on m1 without use of hash function S1 = $(m1)^d$ mod N
   RSA Signature on m2 without use of hash function S2 =$(m2)^d$ mod N

   Someone who has access to signatures on m1 and m2 can produce signature on m1* m2, without access to the key as follows: S1 * S2 = $(m1*m2)^d$ mod N = signature on m1* m2.

   Using a hash function prevents such problems and also reduces computational effort as signing a hash output is less expensive than signing the entire message.

6. When a hash function is used in the generation of digital signatures, what property(ies) of hash function is critical for the security of such signatures?
   Collision resistance. An adversary shouldn't be able to find another message m' with the same hash. Otherwise, he might be able to replace the original message m with m' and the make the receiver believe that the sender sent m'.
   Finding m' such such that h(m') = h(m) for a given m is weak-collision resistance.
   But as discussed in Slide 11 in class on Nov 3, an adversary might be able to pick m and m' both. Therefore, Strong collision resistance comes into play.

7. When using HMAC with a key size of K bits and output of L bits what is i) the security strength against forgery, and ii) security strength of the HMAC algorithm?
   [Hint: See NIST SP 800-107 --
   http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-107r1.pdf]

   i) Strength against forgery – L bits

ii) Strength of HMAC algorithm – K bits.