Final: Part I

Note

This is a closed-book exam. Part I contains five problems, each accounting for 10 points.

Problems

1. A hash code can be used to provide message authentication in various ways. Let H be a hash function. Assume that A and B share a secret key K and a common secret value S. Below are three possible ways to achieve message authentication for message transmissions from A to B:

$$(1) \quad A \rightarrow B\text{: } E(K, [M \parallel H(M)])$$
$$(2) \quad A \rightarrow B\text{: } M \parallel E(K, H(M))$$
$$(3) \quad A \rightarrow B\text{: } M \parallel H(M \parallel S)$$

   (a) In each case, what are the corresponding operations that should be performed by B?

   (b) How do the three schemes compare?

2. Consider a hash function H that uniformly maps b-bit message blocks to n-bit hash values, where $b > n$.

   (a) For any two randomly selected messages of b bits, what is the probability that they have the same hash value? Please show your calculation.

   (b) For any m randomly selected messages of b bits, what is the probability that two of them have the same hash value? Please show your calculation.

3. Describe a scheme (of your choice) for controlling key usage and discuss its advantages and disadvantages.

4. Consider the following communication protocol: each node N in the network has been assigned a unique secret key K_n. This key is used to secure communication between the node and a trusted server C, which stores all the keys. User A, wishing to send a secret message M to user B, initiates the following protocol:

(1) A generates a random number R.
(2) $A \to C$: $A \parallel B \parallel E(K_a, R)$
(3) $C \to A$: $E(K_b, R)$
(4) $A \to B$: $E(R, M) \parallel E(K_b, R)$
(5) B, knowing K_b, decrypts $E(K_b, R)$ to get R and then uses R to decrypt $E(R, M)$ to get M.

An intruder with legal access to one of the network nodes may be able to obtain the plain text of any secret message that has been transmitted between two other nodes. Please describe such an attack.

5. Below is a simple authentication protocol that has been studied in the literature. It relies on a trusted third party C that shares a (distinct) secret key with each principal in the system. The key shared between C and a principal P is denoted K_P. A nonce is essentially some information that has never appeared before (at the time when the nonce is generated).

(1) $A \to B$: "I am A"
(2) B : generate nonce n
(3) $B \to A$: n
(4) $A \to B$: $E(K_A, n)$
(5) $B \to C$: $E(K_B, [A \parallel E(K_A, n)])$
(6) $C \to B$: $E(K_B, n)$

(a) Please explain why A can be certain after step (6) that (assuming there is only one session of this protocol running) it was really A who sent the message "I am A".

(b) The protocol above is in fact incorrect when there can be multiple sessions running simultaneously. Can you find a problematic scenario?