CSCE 44303/54203 Cryptography – Homework 6

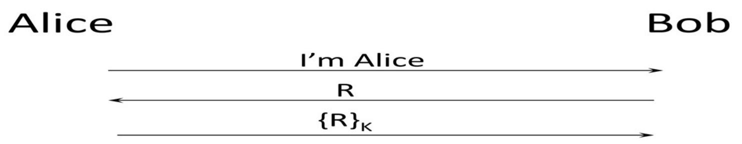
Release date: November 5, 2024

Due date: November 12, 2024

Full Grade: 108 pts

1. **(12 pts)** Why is it not good to directly store the hash of password in a file? Why can the use of salt enhance password security? Can the use of salt prevent an attacker who got the salted hash of a password from inferring the password?
   a. **If a hash of a password is stored directly in a file, then attackers can still use methods to reverse engineer the original password, such as Dictionary attacks.**
   b. **With salt enhanced password security this helps to prevent attacks such as the Dictionary attack because no two hashes will have the same hash. If Bob and Alice have the same password, the stored hash will be different.**
   c. **No, Salt alone does not prevent an attacker from inferring the password. Salt makes it harder to do "hash comparison" and dictionary attacks. However, if the attacker has the salted hash and salt value, brute force attacks are still viable.**

2. (12 pts) Suppose the following challenge-response protocol is used by two parties which share a secret K to mutually authenticate each other:



The figure shows how Alice authenticates herself to Bob, but Bob can authenticate himself to Alice in a similar way (i.e., exchanging the roles of Alice and Bob in the figure). After authentication, the two parties want to set up a session key. Which of the following are secure to be used as a session key? For those not secure, explain why.

1) K XOR R;
   a. Not secure, if an attacker captures multiple session keys, they might be able to infer information about K due to the way XOR functions. Since R is a random value known to both parties, an attacker who observes K XOR R might be able to reverse engineer K with enough observations or if R is reused.
2) $E_K(R+K)$;
   a. Secure, provides non-reversible session key that incorporates both K and R in an encrypted form.
3) $E_K(K+1)$;
   a. Not secure, does not incorporate the randomness of R making it to predictable and vulnerable to replay attacks on K.
4) $E_{K+R}(R)$
   a. Secure, session key is tied to both K and R allowing for each session to be unique

2. **(8 pts)** Design a two-message authentication protocol, assuming that Alice and Bob know each other's public keys, which accomplishes both mutual authentication and establishment of a

session key. Note that Alice can only send one message to Bob and Bob can only return one message back.

a. **Alice Generates a nonce N_A**
b. **Alice generates a random session key K_{AB}**
c. **Alice sends the following message to Bob**
    i. **$M_1 = E_{PKB}(K_{AB}||N_A||S_{SKA}(K_{AB}||N_A))$**
    ii. **$K_{AB}||N_A$ contains the session key and nonce, both encrypted with Bob's public key.**
    iii. **$S_{SKA}(K_{AB}||N_A)$ is Alice's signature on the session key and nonce, allowing for Bob to verify the message came from Alice.**
    iv. **All of this is encrypted with E_{PKB} or Encrypt with Bobs public key**
d. **Bob Receives M_1 and decrypts it using his private key SK_B to retrieve K_{AB}, N_A, and $S_{SKA}(K_{AB}||N_A)$.**
e. **Bob verifies $S_{SKA}(K_{AB}||N_A)$ by using Alice's public key PK_A and confirms the message was sent by Alice, and has not been tampered with.**
f. **Bob generates a nonce N_B**
g. **Bob Sends the following response message to alice**
    i. **$M_2=E_{PKA}(N_B||N_A||S_{SKB}(K_{AB}||N_B||N_A))$**
    ii. **$N_B||N_A$ is Bob's nonce and Alices nonce encrypted with Alice's public key.**
    iii. **$S_{SKB}(K_{AB}||N_B||N_A)$ is Bob's signature on the session key, his nonce, and Alices nonce. Allowing Alice to confirm the message came from Bob and agree on session key K_{AB}**
h. **Alice Receives M_2 and decrypts using SK_A to retrieve N_B, N_A, and $S_{SKB}(K_{AB}||N_B||N_A)$**
i. **Alice Verifies the signature $S_{SKB}(K_{AB}||N_B||N_A)$ using Bobs public key PK_B and if valid then The message is from Bob, Bob has received K_{AB} and acknowledges N_A, and now both Alice and Bob authenticated each other and agreed upon a session key K_{AB}**

3. (6 pts) Suppose a student needs to login a public workstation in a computer lab, and access several different services provided by the university. Considering this scenario, what is the main motivation of using Kerberos system?

a. **The main motivation of the Kerberos system is to allow the student to log in once and gain access to multiple services without having to re-enter a password for each service. Also it utilizes a time ticketed system which allows for "passwordless" sign-on for a certain period of time. After the time expires, then a password will be required. This reduces the risk of unauthorized access to unattended workstations. Kerberos does not send passwords over the network, helping to prevent interception.**

4. (6 pts) In password-based online authentication, why can't a user directly send the hash of his/her password to the server to get authenticated (assuming the server knows the user's password)?

a. **If the hash is intercepted then the attacker can simply resend the hash (sometime in the future) to impersonate the user, without ever needing the unhashed password.**

5. (16 pts) Suppose Merkle hash tree is used to authenticate eight messages, m1, m2, …, m8. The eight messages are used to build the Merkle tree, and the root of the tree is signed with a digital signature s. i) To authenticate m3, besides m3 itself and the signature s of the tree root, which elements should be included in the message? How can the receiver verify the integrity of m3? ii) To authenticate m8, besides m8 itself and the signature s of the tree root, which elements should be included in the message? How can the receiver verify the integrity of m8?

a. **Elements needed for m3 besides m3 and the signature s**
    i. **Hash(m4), Hash(m1), and Hash(m2)**
    ii. **The hash of the pair generated by hashing Hash(m5) through Hash(m8)**
b. **Verification process**
    i. **Receiver computes Hash(m3)**
    ii. **Receiver combines Hash(m3) with Hash(m4) making parent hash m3 and m4**
    iii. **Combined hash is merged with Hash(m1) and Hash(m2)**
    iv. **Finally the computed Combined hash is then merged with hash value of m5 – m8 to compute the root hash.**
    v. **If computed root hash matches the signed root hash, receiver trusts the integrity.**
c. **Elements needed for m8 besides m8 and signature s**
    i. **Hash(m7)**
    ii. **Hash pair of Hash(m5) and Hash(m6)**
    iii. **Hash pair of hashes m1 – m4.**
d. **Verification Process**
    i. **Receiver computes Hash(m8)**
    ii. **Receiver combines Hash(m8) with Hash (m7)**
    iii. **Receiver merges Hash with the combined hash of m5 and m6**
    iv. **Receiver merges Hash with the hash of m1 – m4 to compute root hash**
    v. **If computed root hash matches the signed root hash, receiver trusts the integrity.**

7. (12 pts) Suppose user A is broadcasting packets to n recipients B1, …, Bn. Privacy is not important but integrity is. In other words, each of B1, …, Bn should be assured that the packets he is receiving were sent by A. User A decides to use MAC.
a. Suppose user A and B1, …, Bn all share a secret key k. User A MACs every packet she sends using k. Each user Bi can then verify the MAC. Explain why this scheme is insecure, namely, show that user B1 is not assured that packets he is receiving are from A.
b. Suppose user A has a set S = {k1, …, km} of m secret keys. Each user Bi has some subset Si of the keys. When A transmits a packet she appends m MACs to it by MACing the packet with each of her m keys. When user Bi receives a packet he accepts it as valid only if all MAC's corresponding to keys in Si are valid. What property should the sets S1, …, Sn satisfy so that the attack from part (a) does not apply? We are assuming all users B1, …, Bn are sufficiently far apart so that they cannot collude.
c. Show that when n = 10 (i.e. ten recipients) the broadcaster A need only append 5 MAC's to every packet to satisfy the condition of part (b). Describe the subsets S1, …, S10 of set {k1, …, k5} you would use.

a) All recipeints share the same secret key k and any recipient can create a valid MAC using k for a packet and then broadcast it to others. This creates a scenario where B1 has no way of knowing whether the packet actually came from A or if it was generated and sent by another recipient. This leads to a failure in verifying the authenticity of the senders of packets. Allowing anyone to "forge" messages.
b) The main required property of Sets S1 – Sn is that the subsets should be constructed so that NO recipient Bi, has the same subset of keys as any other recipient.
c) Design
    a. S1 = (k1,k2,k3)
    b. S2 = (k1,k2,k4)

c. S3 = (k1,k2,k5)
d. S4 = (k1,k3,k4)
e. S5 = (k1,k3,k5)
f. S6 = (k1,k4,k5)
g. S7 = (k2,k3,k4)
h. S8 = (k2,k3,k5)
i. S9 = (k2,k4,k5)
j. S10 = (k3,k4,k5)

8. (16 pts) Revise the Onion Routing protocol presented in slides 5-6 of Module 12 to have one new protocol that meets two requirements: (i) there are only two relays N1 and N2; (ii) N2 is permitted to know the content of the request m. Give the request sending process and the data reply process.

SENDING

a) Client sends the following $Enc_{N1}(Enc_{N2}(m)||Address\ of\ N2)$ to N1
   a. $Enc_{N2}(m)||$Address of N2
      i. Encrypt m with public N2 key
      ii. Attach the Address of N2 to the sent packet
   b. $Enc_{N1}(*)$ Encrypt With Pubic N1
b) N1 receives the packet and decrypts the received packet using its private key
c) N1 obtains $Enc_{N2}(m)$ and Address of N2
d) Using the provided Address for N2, N1 sends $Enc_{N2}(m)$ to N2.
e) N2 decrypts the Message m and forwards it to the destination If needed.
   a. If there is a receiver then N2 would most likely encrypt m with receivers public key, then send it to the receiver for them to decrypt with there private key.

REPLY

a) N2 generates a reply message r for the Client
b) N2 encrypts r with public key N1 $Enc_{N1}(r)$ and sends this to N1
c) N1 decrypts the reply packet using its private key, obtaining r
d) N1 encrypts r with the Client's public key $Enc_{CLIENT}(r)$ and transmits this to the Client
e) Client decrypts the received message and is able to read the reply

9. (8 pts) Consider homomorphic encryption systems. Suppose c1 and c2 are the ciphertexts of two numbers m1 and m2 under the RSA homomorphic crypto system. If you decrypt the product of c1 and c2, what will you get? If c1 and c2 were encrypted by AES, then can you get anything meaningful by decrypting the product of c1 and c2?

RSA

a) Assume $c1=Enc_{RSA}(m1)$ and $c2 = Enc_{RSA}(m2)$
b) $c = m^e \bmod n$
c) $c1 = m1^e \bmod n$
d) $c2 = m2^e \bmod n$
e) $c1 * c2 = (m1^e) * (m2^e) \bmod n = (m1 * m2)^e \bmod n$
f) Decryption of c1 * c2 will yield m1 * m2.
g) Therefore in RSA, decrypting c1 * c2 will give a meaningful result in the form of m1 * m2.

AES

a) Assume c1=Enc_{AES}(m1) and c2 = Enc_{AES}(m2)
b) Decrypting the product of c1 and c2 will not yield any meaningful results due to the structure of AES encryption which does not allow for any correspondence between product of two ciphertexts.

10. (12 points) Suppose an aggregator wants to know the sum of *n* users' ages. Design a scheme which allows the aggregator to obtain the sum without revealing any individual user's age to any other party. Assumptions about the system: users can communicate to each other; no party is trusted; at most three users can collude; no user colludes with the aggregator; no party can eavesdrop the communications between any two users. You must design your own solution. Do not copy from any publication.

SOLUTION

a) Each user will generate n-1 random numbers r_{ij} with j = 1,2,…n and j != i
b) $R_{i1} + R_{i2} + \ldots + R_{in} = 0$ making all random numbers summed, cancel out
c) If $j < i$, user U_i will send r_{ij}, to user U_j
d) If $j > i$, user U_i will receive r_{ji}, to user U_j
e) Next each user calculates there masked age mi by adding there age ai to all the random values they generated and received.
    a. $M_i = a_i + \sum^{n}_{j=1} r_{ij}$
f) Next the aggregator Adds up all values computed for M_i leading to the sum of n users being given.

EXAMPLE

a) Assume there are only 3 users in this example with U1,U2,and U3 with ages 25,30,and 35.
b) The sum of there ages should equal 90
c) First U1 generates $r_{12} = 3$ and $r_{13} = -2$
d) U2 generates $r_{21} = -3$ and $r_{23} = 4$
e) U3 generates $r_{31} = 2$ and $r_{32} = -4$
f) When all rs are added up they sum to 0
    a. 3-2-3+4+2-4 = 0
g) Next each user calculates there masked age
h) U1 calculates m1 = age1 + r_{12} + r_{13} = 26
i) Each U calculates mask with the same formula
    a. M2 = 31
    b. M3 = 33
j) The aggregator adds M1 + M2 + M3 = 90 giving the sum of the three ages without reveling the ages to the aggregator.