Ryan Young

CSE 3400

4/8/2020

Homework 7

1.) *Design a simple and efficient protocol allow a set of users {1, n} to exchange message securely among them, i.e. ensuring confidentiality, authentication, and integrity of messages. Your design may assume that users know the public keys of each other; furthermore, you may assume that each user i has two pairs of public-private keys: an encryption-decryption key-pair (e(i), d(i)) and a singing-verifying key-pair (s(i), v(i)) of every user I, are known to all users. Your descriptions should be very clear, well-defined and concise, but can be high-level, e.g., use (a, b, c) <- α to denote "splitting" a tuple α = (a, b, c) into its components. Please read all three below parts before answering; the third part is an additional requirement that you may meet already in answer of parts 1 and 2, this may save time or answering part 3.*

1.) *A send proves that receives message m, receiver identifier I, a sender identifier j and the sender's private keys d(j), s(j), and produces a ciphertext c to be sent to I, I.e. c = sendj, d(j), s(j)(m, j).*

My send function would utilize the fact that each user in the set knows the other user's public keys. So, I would use a sign then hash with a hash then MAC then encrypt scheme. The signing would be done using the sender's private key this way any user who the send can send to will be able to verify that It was the right person who sent it. Then I would use the public key of the receiving user in order to encrypt the message this will allow only the right receiving user to decrypt the message and I will use the public key of the sender for the MAC. I would use RSA for both the signing and verifying as well as the encryption and decryption. The signing process provides authentication, the RSA encryption provides confidentiality and the MAC provides the integrity.

*2.) A receive process that receives ciphertext c, receiver identifier I, a sender identifier j and the receivers private keys d(i), s(i), and produces message m and sender identifier j, if c was output of send executed by j on message m with receiver I, and an error indicator otherwise.*

My receive process would obviously mirror that of my send process. A user would receive a message and then they would use the sender's public key to verify it was the correct sender and the message was unchanged using the MAC. Then the receiving process would use the receiver's private key in order to decrypt the ciphertext into the message and boom we have secure transmission of messages that provides, Confidentiality, Authentication, and integrity.

*2.) The RSA algorithm calls for selecting e and then computing d to be its inverse (mod Φ(n)). Explain how the key owner can efficiently compute d, and why the attacker cannot do the same.*

A key owner can efficiently compute d and an attacker cannot do the same because the key owner should know the factors of n which would be p and q. When computing (mod Φ(n)) you would need these factors p and q and since finding factors of large numbers is a computationally hard problem the attacker cannot compute d efficiently.

3.) *Consider an application that caches users' name and e in their public key (but not n). Suppose Alice's public key is ($e_A$, $n_A$) and the application includes sender's username and the complete pubic key (e, n). Show how an attacker can trick the application into believing - incorrectly - that an incoming message sent by the attacker was signed by Alice.*
Since n is not cached by the application and $e_A$ is Alice's public key the attacker will know $e_A$ and can choose any p and a q such that they make their own n value that will work easiest for them. This will allow the attacker to make their own p, q, n and d thus giving the attacker everything they need in order to fake out the sever and pretend to be Alice. They would be able to send any message they want because they know e so $m^{eA}$ mod n would appear as it has come from Alice and because they know D they should be able to decrypt anything as well.

4.) *You are given textbook-RSA ciphertext c = 281, with public key e = 7 and modulus (n) = 341. Compute the private key d and the message $m=c^d$ mod n.*
We know that $d = e^{-1} = 1$ mod Φ(n). Therefore, we need to find the factors of 341 which are 31 and 11. Now we know that Φ(n) = (p-1)(q-1) = (31-1)(11-1) = 300. We can now modify the equation so, 7d mod 300 = 1. So, we just need to find a d such that it is one greater than 300 which happens to be 43. Now since d = 43 m = $281^{43}$ mod 341. Since this number $281^{43}$ is a huge number we need to break it down. The factors of 43 are 8 and 5 so this becomes (($281^5$ mod $314^8$ mod 341 * ($281^3$ mod 341)) mod 341. $281^5$ mod 341 -> $281^3$ mod 341 * $281^2$ mod 341. We can calculate that $281^2$ mod 341 = 190 using this we can simplify and get ($32^8$ mod 341) * 194) mod 341 -> 1 * 194 mod 341 = 194. Therefore m = 194 and d = 43.