CSCI 6634/4534 – Cryptography – HW 3 – 24 Spring – Due (on Canvas): 4/19/24 Friday 11.00 pm

- Quiz 3 **in-person, open book, open notes** will be on 4/25/24 Thursday and will cover these topics:

  AUTHENTICATION APPLICATIONS: Kerberos version 4, X.509 certificates.
  DIGITAL SIGNATURES: DSA.
  EMAIL SECURITY: Pretty Good Privacy.
  WEB SECURITY: Secure Socket Layer.

1. (15 points) This question is about Kerberos Version 4 from Table 15.1. Give explanations for the following:

   (a) Suppose BG captures (4) i.e. the communication from TGS to C. Will he be able to pose as C and get service from V?

   (b) Explain what would be the disadvantage if the lifetime in the service-granting ticket was very long and what would be the disadvantage if the lifetime was very short.

   (c) Suppose V is a file server and C wants to get files from V. One purpose of sending $K_{C,V}$ in round (4) is that V can encrypt files using $K_{C,V}$ to mantain confidentiality. Suppose C does not care about the confidentiality of these files i.e. he does not care if BG is able to read these files. In this situation, the files do not need to be encrypted with $K_{C,V}$. In this situation, does $K_{C,V}$ still need to be included in round (4)? Give a YES/NO answer and then briefly explain your answer. Give a YES/NO answer and then briefly explain your answer.

2. (15 points)

   In this problem we look at the question of why the value of k should not be repeated in DSA. BG is the cryptanalyst; he has access to all information except for k and x. Suppose A, the person signing the message, has used the same value of k twice as follows: A first signs a message $H(M_1)$ to get a signature r_1, s_1. Then A, using the same k value, signs a message $H(M_2)$ to get a signature r_2, s_2.

   Show what BG would do by showing how BG's attack would work on the following example. BG knows that $q = 11$ (the values of p, g, r are not relevant to this problem).

   For the first message signed by A, $H(M_1) = 5, s_1 = 3$.

   For the second message signed by A, $H(M_2) = 2, s_2 = 4$.

   The message which BG wants to forge has $H(M_3) = 3$. What will be the s_3 part of the signature of this message? Show your work

   Hint: Think about what information can BG get by considering the equations $s_1k = [H(M_1) + xr_1] \bmod q$ and $s_2k = [H(M_2) + xr_2] \bmod q$, and the fact that r does not depend on the message i.e. $r_1 = r_2$. Please note that in order to forge a signature for the fake message M3, the BG does not actually need to figure out x, it is enough for him to figure out k and $x * r$.

3. (25 points) Give brief explanations for the following:

   (a) We have seen that to stop replay attack, either time stamps or nonces may be used. What are the relative advantages of each i.e. explain one advantage nonces have over time stamps and one advantage time stamps have over nonces.

   (b) In PGP, if a user has two different public keys generated at random, calculate the probability that these two keys will have the same key ID?

   (c) Explain in brief how PGP stores a user's private key.

   (d) Suppose SSL is being used in your browser when you log in to your bank's website. Assume that the SSL handshake protocol has been completed and now, before you log in to your bank account, it asks you for a password. Now suppose that the bad guy is eavesdropping - will the BG be able to figure out what the password is and log in as you next time? Give a YES/NO answer and then briefly explain your answer.

   (e) Briefly explain how SSL stops replay attacks i.e. what prevents replay of earlier SSL handshake messages.

4. (10 points) Suppose messages are numbers between 1 and 1000. Consider the hash function $h(M) = ((M \bmod 91) * 10) \bmod 73$. So, for example, $h(99) = ((99 \bmod 91) * 10) \bmod 73 = 80 \bmod 73 = 7$. Suppose a digital signature scheme worked as follows. If A wanted to sign a message M, his signature would be $E_{KR_A}(h(M))$, where the encryption is being done with RSA. Is this a good scheme?

   (a) Give a YES/NO answer.

   (b) Explain your answer i.e. if you said YES explain why you think this is a good scheme, if you said NO explain why by showing that if you were the bad guy BG and intercepted the message $M, E_{KR_A}(h(M))$, how you would convince B that another message M' (which you as the BG have generated and which is different from M) had actually been signed by A. Illustrate your answer by showing that if $M = 103$, what would be another number M' which you could persuade B to accept.

Programming Problem (35 points): What you have to turn in and where you have to turn it in is similar to how you needed to turn in stuff for the earlier HW including the self-critique, the easy to read source code, a printout of the output in the main HW3 submission (along with all the other HW3 stuff, all of this to be turned in as a single pdf file) and the actual program in the Programming Problem submission.
You have to write a program to implement DSA. Your program will take as input p, q, h, x, k, a "real" value $H(M_1)$, and a "fake" value $H(M_2)$ (which will be different from $H(M_1)$). Your program should

1. first calculate g and y and then print these values.
2. then calculate (for $H(M_1)$) the signature (r, s) and then print these values.
3. then make sure that DSA works correctly on real values by verifying that (r, s) is the signature for $H(M_1)$ by calculating w, u_1, u_2, v and testing whether $v = r$? Your program should print w, u_1, u_2, v and the result of the test.
4. then make sure that DSA works correctly on fake values by verifying that (r, s) (use the same r, s values as you calculated for $H(M_1)$) is not the signature for $H(M_2)$ by calculating w, u_1, u_2, v and testing whether $v = r$? Your program should print w, u_1, u_2, v and the result of the test. Here what you are making sure of is that if BG tries to forge a signature of a new message by using an old signature, this should not work.

You have to run your program on three different inputs.

1. the example discussed in class and in the handout: $p = 7, q = 3, h = 3, x = 2, k = 1, H(M_1) = 3, H(M_2) = 4$.
2. $p = 47, q = 23, h = 5, x = 7, k = 13, H(M_1) = 5, H(M_2) = 4$.
3. An input generated by you - you should generate the largest values for which you can get correct results in a "reasonable" amount of time.

Extra Credit Problems: 13.1, 13.7, 14.2, 14.3, 14.6, 15.2, 17.1, 19.7.
Extra Credit Problem 4:

1. Design and implement your own simple 20 bit hash function.
2. Implement a birthday attack on this hash function.

As usual, you should a printout of the program listing, a printout of the output, and submit the soft copy on blackboard.
Extra Credit Problem 5:
Find a hash function which has the weak collision resistance property but not the strong collision resistance property.

1. Show what your function is.
2. Explain why your function has the weak collision resistance property.
3. Explain why your function does not have the strong collision resistance property.

Extra Credit Problem 6: Implement a brute force attack on DSA and show how it works on some examples. As usual, you should a printout of the program listing, a printout of the output, and submit the soft copy on blackboard.