CSCI 667 - Homework 3*

Assigned October 3rd, 2024; Due 11:59pm on October 17th, 2024

Prof. Adwait Nadkarni

Note: This homework assignment is worth 70 points.

1 RSA {30 points}

(a) {5 points} Prof. Pedantic generates an RSA keypair, consisting of a public key $\langle e, n\rangle$ and a private key $\langle d, n\rangle$. He saves the large (2048-bit) prime factors p and q used to compute n (i.e., $n = pq$) as well as his public key $\langle e, n\rangle$. Ever forgetful, he forgets to save his private key $\langle d, n\rangle$. D'oh! Will Prof. Pedantic be able to recover and re-generate his private key? That is, can he learn d given the information that he recorded? Why or why not?

(b) {10 points} Prof. Pedantic decides to create a variant of RSA, which he calls RASP (the P is for Pedantic). In RASP, the private key exponent d is computed as $d = e^{-1} \bmod n$ (the modular inverse of the public key exponent e, modulo n), as opposed to $d = e^{-1} \bmod \Phi(n)$. The encryption and decryption functions also differ between RSA and RASP, but knowing what these functions are isn't important for this question.

What **major** security flaw does RASP's key generation algorithm introduce?[1]

(c) {5 points} Prof. Pedantic gives you (securely) his RSA public key:

$$K^+ = \langle e = 13, n = 77\rangle$$

What is the corresponding ciphertext for the plaintext message $M = 2$, encrypted with Prof. Pedantic's public key? Show your work.

(d) {5 points} Given his public key $\langle e = 13, n = 77\rangle$, what is Prof. Pedantic's private key?

(e) {5 points} Why were you able to answer the previous question?!? That is, is RSA broken? If it isn't broken, then why were you able to derive his private key from his public key?

*Last revised on October 3, 2024.

[1]Note that the notation $a = b^{-1} \bmod q$ is equivalent to $ab \bmod q = 1$. Both reflect the fact that a and b are modular inverses under modulo q.

2 *PayMe!* {10 points}

Hoping to become the next dotcom millionaire, Prof. Pedantic decides to create an online money payment service similar to PayPal. His service, *PayMe!*, allows users to transfer money to other users of the system.

To ensure that no fraudulent activity takes places, the *PayMe!* service stores the public key of each user. (You should assume that the sharing of the public key is secure; that is, the server has each user's correct public key.)

If Alice ("A") wishes to give X dollars to Bob ("B"), she sends the following message to the *PayMe!* service ("S"):

$$A \rightarrow S : A, B, X, n, \{X|n\}_{A^-}$$

where n is a nonce, A^- is Alice's private key, and $\{M\}_{K^-}$ denotes[2] a digital signature over M computed using the private key K^-.

(a) {5 points} What is a nonce, and why does Prof. Pedantic include one in his protocol? Does it prevent any type of attack?

(b) {5 points} Explain how an active adversary can exploit a weakness in Prof. Pedantic's protocol to steal money from an honest user, Alice.

3 PompousPass$^{\text{TM}}$ {15 points}

Prof. Pedantic believes he has a come up with a simple way of performing authentication. His system, PompousPass$^{\text{TM}}$, uses RSA signatures. Let Id_x and Pw_x respectively be the username and password for user x, and (x^+, x^-) be the public/private keypair associated with user x. Assume that the server S knows the user's username (Id_x), password (Pw_x), and his public key (x^+). To authenticate to the server, x sends:

$$x \rightarrow S : Id_x, r, \{Id_x|Pw_x|r\}_{x-}$$

where r is a nonce.

The server decrypts the message using x's public key, and authentication proceeds iff (1) the transmitted password matches the password stored in the server's database and (2) the nonce is fresh. Note that this is the only message in the authentication protocol.

(a) {4 points} Is this system secure? Why or why not?

(b) {7 points} Assuming that the parties share no keys other than their public keys, provide a revised protocol that better protects the client's password.

[2] Yikes! Another notation for signatures! There are several ways to denote the public key signature operation, including these three which appear frequently in the literature: $E_{K^-}(M) = \{M\}_{K^-} = \sigma_{K^-}(M)$. All three notations are equivalent.

(c) {4 points} In practice, public-key cryptography is often used to distribute session keys, which are then used with symmetric algorithms. Why is this approach preferred over using solely public-key operations? (1-2 sentences)

4 Kerberos-ish {15 points}

Dissatisfied with Kerberos (he's difficult to please), Prof. Pedantic proposes a simpler protocol called WoofWoof$^{\text{TM}}$ that eliminates the use of the TGS. Let ID_C and IP_C be the respective ID and IP address of the client, ID_V be the ID of the service that the client wishes to access, K_C be a pre-shared key between the KDC and client, K_V be a pre-shared key between the KDC and service V, $K_{C,V}$ be a temporary key generated by the KDC during the course of the protocol, $E_X(M)$ be the encryption of M using key X, and *lifetime* be the lifetime of a ticket. Finally, let $A \rightarrow B : M$ denote the transmission of message M from A to B.

Prof. Pedantic's protocol works as follows:

(i) Client $\rightarrow$ KDC: ID_C, ID_V

(ii) KDC $\rightarrow$ client: $E_{K_C}(K_{C,V}, \text{lifetime}, \text{ticket}_V)$, where $\text{ticket}_V = E_{K_V}(K_{C,V}, \text{ID}_C, \text{IP}_C, \text{lifetime})$.

(iii) Client $\rightarrow$ service V: $\text{ticket}_V, E_{K_{C,V}}(\text{ID}_C, \text{IP}_C, t)$, where t is the current time.

(iv) Service V $\rightarrow$ client: $E_{K_{C,V}}(t+1)$.

(a) {5 points} Does WoofWoof$^{\text{TM}}$ achieve client authentication? How (if yes) or why not (if no)?

(b) {5 points} Does WoofWoof$^{\text{TM}}$ achieve server authentication? How (if yes) or why not (if no)?

(c) {5 points} By removing the TGS, what key functional goal of Kerberos does WoofWoof$^{\text{TM}}$ **not** achieve?

Hint: For parts (a) and (b), use the reasoning found on the Kerberos class slides for arguing for or against authentication.

Submission Instructions

Submit your solution as a single PDF using Blackboard. To upload your assignment, navigate to the "Concepts of Computer Security (Fall 2024) course. Use the "Homework 3" assignment.

Writeups submitted in Word, ASCII, PowerPoint, Corel, RTF, Pages, and other non-PDF formats will not be accepted. Consider using LaTeX to format your homework solutions. (For a good primer on LaTeX, see the Not So Short Introduction to LATEX.)

Note that you may submit a PDF scan of hand a hand written solution; however, you will receive **0 points** if the instructor cannot read your hand writing. If the instructor has any difficulty reading your hand writing, you may not submit hand written solutions for future assignments.

Please post questions (especially requests for clarification) about this homework to Piazza.