CSC474 - Homework 3 Written Questions (Part 1)*

Assigned October 4th, 2022; Due 11:59pm on Tuesday October 18th, 2022

45 points

Prof. William Enck

1 Part 1: Written Questions

1.1 *PayMe!* {10 points}

Hoping to become the next dotcom millionaire, Bob 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"):

$$A \rightarrow S : A, B, X, n, Sig(A^-, [X|n])$$

where n is a nonce, A^- is Alice's private key, and $Sig(K^-, M)$ denotes a digital signature over M computed using the private key K^-.

(a) {5 points} What is a nonce, and why does Bob 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 the *PayMe!* protocol to steal money from an honest user, Alice.

1.2 PompousPass {20 points}

Bob believes he has a come up with a simple way of performing authentication. His system, PompousPass^{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

*Last revised on October 2, 2022.

server S knows the user's username (Id_x), password (Pw_x), and his public key (x^+). To authenticate to the server, x sends:

$$x \rightarrow S : Id_x, r, Sig(x^-, [Id_x|Pw_x|r])$$

where r is a nonce.

The server should only authenticate the user iff (1) the transmitted password matches the password stored in the server's database and (2) the nonce is fresh.

(a) {5 points} Describe two attacks on this protocol. You should assume the attacker cannot get access to the password database.

(b) {10 points} Fix the protocol to defend against these attacks.

(c) {5 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)

1.3 Kerberos-ish {15 points}

Dissatisfied with Kerberos (he's difficult to please), Bob proposes a simpler protocol called WoofWoof™ 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, $K_{C\text{-}V}$ be a temporary key generated by the KDC during the course of the protocol, $E(K_X, M)$ be the encryption of M using key K_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.

Bob's protocol works as follows:

$$\begin{aligned}
\text{Client} \rightarrow \text{KDC} &: \text{ID}_C, \text{ID}_V \\
\text{KDC} \rightarrow \text{client} &: E(K_C, [K_{C\text{-}V}|\text{lifetime}|\text{ticket}_V]), \text{ where ticket}_V = E(K_V, [K_{C\text{-}V}|\text{ID}_C|\text{IP}_C|\text{lifetime}]) \\
\text{Client} \rightarrow \text{service } V &: \text{ticket}_V, E(K_{C\text{-}V}, [\text{ID}_C|\text{IP}_C|t]), \text{ where } t \text{ is the current time} \\
\text{Service } V \rightarrow \text{client} &: E(K_{C\text{-}V}, [t+1])
\end{aligned}$$

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

(b) {5 points} Does WoofWoof™ 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™ **not** achieve?