CSI 4139 / CEG 4399
Design of Secure Computer Systems

Laboratory #2
Due: October 4, 2018 (written report due October 9, 2018)

Goal: Use 2-factor authentication and various cryptographic algorithms to establish a secure channel with a host site.

Details: Create a website that a user logs into using both a keyboard and a cell phone. Once the user clicks the login button / link at the website, the user is presented with a login screen. When the user enters his/her username and hits <return>, the website consults a file that it keeps locally to see which cell phone number is associated with that username. The website then sends an SMS text message to that phone number. The user, having received the text message, enters it and a password to a local application, which does some processing and returns a string. The user enters this string at the login screen and hits <return>. The website validates the sent value; if correct, it returns 2 strings which the user can input to the local application. The application validates these strings and returns *success* or *fail* to the user.

During the lab demonstration of your implementation, the TA will confirm the successful operation of your program.

Deliverables: You are expected to write a document and demonstrate your program.

Document: Write a brief description (no more than one page) of your program. In particular, you should describe your programming environment, programming language, and any underlying assumptions about the communications environment. Along with this, you must include a detailed analysis (approximately two pages) of your implementation. (You will need to choose encryption, MAC, and signature algorithms, and appropriate key and parameter sizes to balance security, efficiency, and usability. You must discuss and defend all your choices.) Finally, include approximately one page discussing any disadvantages that you see with this protocol.

Software: You must implement the software to do the protocol described on the following page. This will consist primarily of code to send and receive the appropriate messages, as well as a user interface sufficient to allow the TA to observe what your program is doing. (*Note: if you wish, you can do everything on Alice's side on her phone, rather than on her computer & phone.*)

This laboratory project is to be done in groups of up to 3 people.

Protocol Description

System parameters (known to everyone) are a prime number p and a generator g of $\mathbb{Z}_p^*$. At initialization time, Alice chooses password a, computes $\alpha = g^a \bmod p$, and gives α to the host, who stores it along with Alice's username and cell phone number. The host gives its signature verification key to Alice, who stores this key as a trust anchor.

At login time:

- Alice sends her username ("Alice") to the host, Bob.
- Bob looks up α, chooses (at random) a value b, and computes $\beta = g^b \bmod p$. Bob sends β to Alice's cell phone as an SMS text message.
- At her local application, Alice enters β and her password a. The application computes $K = \beta^a \bmod p = g^{ab} \bmod p$. The first n bits of K are k_1; the next n bits of K are k_2. Let $m = \alpha \,||\, \beta$. The application computes $m_1 = E_{k2}(m \,||\, \mathrm{MAC}_{k1}(m))$ and displays the result to Alice. Alice sends this value to the host through the login screen.
- Bob computes $K = \alpha^b \bmod p = g^{ab} \bmod p$. Bob then uses k_1 and k_2 to validate m_1, constructs $m' = \beta \,||\, \alpha$, computes $m_2 = E_{k2}(m' \,||\, \mathrm{MAC}_{k1}(m'))$, digitally signs m_2 using its private key, and sends $\{m_2, \mathrm{sig}(m_2)\}$ to Alice's login screen.
- Alice enters $\{m_2, \mathrm{sig}(m_2)\}$ to her local application, which validates m_2, verifies the signature, and returns *success* or *fail*.

After the successful completion of this protocol, for all subsequent data traffic between Alice and the host, all messages can be MAC'ed using key k_1 (for authenticity) and all sensitive messages can be encrypted using key k_2 (for confidentiality).

This protocol provides mutual authentication between Alice and the host, Bob (using 2-factor authentication for Alice), key establishment, key confirmation, and secure channel establishment. Alice only needs to remember her password, and there is no need for a trusted third party (e.g., a Certification Authority, CA).