Vincent Yasi

CS 370 -Intro to Security-

Problem Set 7

Multi-Factor Authentication?

Q1 [5 pts]: What is multi-factor authentication? Give a real-world example of its use.

Multi-factor authentication is where more than one form identification or authentication is used. Often it includes using a password and then a second form, such as a one-time use verification code sent to your phone or a piece of biometrics.

A prime real-world use is logging into the OSU website. Most of our logins now use two-factor authentication where we first put in our username and password, and then need to complete a second step on the Duo app to verify.

Q2 [5 pts]: Name four factors of authentication and provide an example for each one.

What entity knows- for example, a password or PIN

What entity has- for example, a keycard or a smart phone with an app

What entity is- for example, fingerprints or retinal scans

What entity does- for example, how they walk or how they sign their name

Q3 [5 pts]: What is the difference between multi-factor authentication and mutual authentication (please look the latter up)?

Multi-factor authentication authenticates a single entity through multiple means. Mutual authentication authenticates multiple entities through mutual means, like exchange of secret keys, etc. The first is many layers for one, while the second is one layer for many.

Biometrics

Q4 [5 pts]: What is a biometric? Give four examples of biometrics used for authentication.

A biometric is some factor or property inherent within an entity itself, often something biological. It is typically relatively unchanging and unique to the entity.

Some examples are fingerprints, retinal scans, face recognition, or handwriting.

Q5 [4 pts]: What is the difference between static and dynamic biometric? Give two examples of each.

A static biometric is one which does not change (or changes very slowly) and so remains a constant biometric that can be used. Examples include fingerprints or retinal scans.

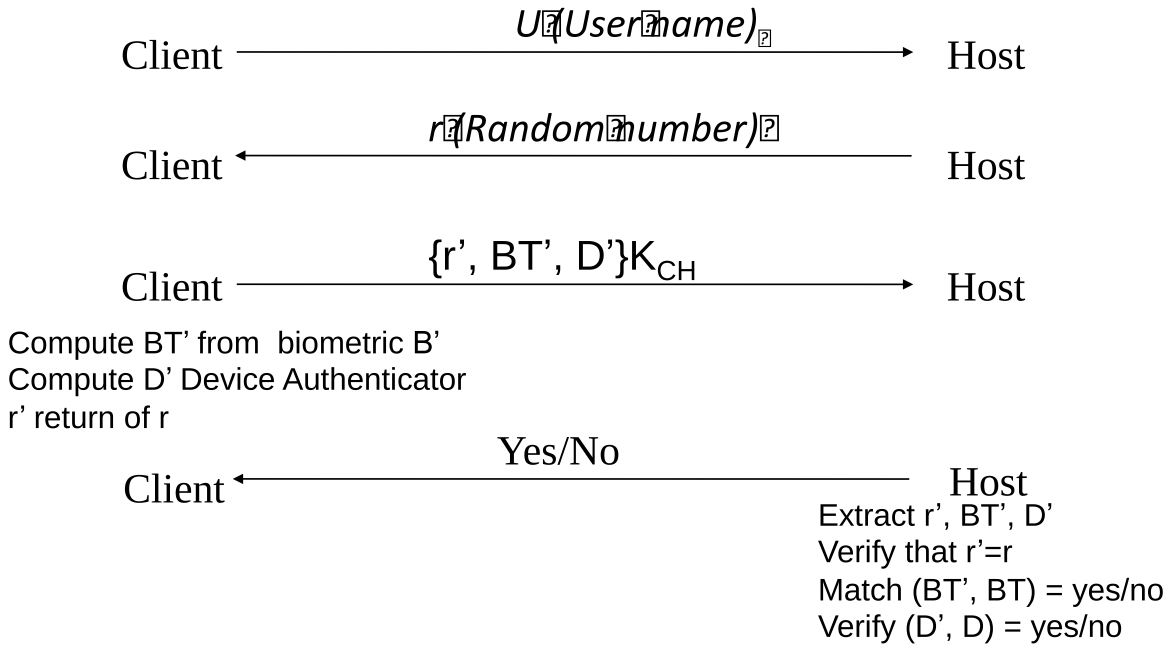
A dynamic biometric is once which is much more prone to changing or is not very concrete and is therefore less likely to be used for authentication. Examples include one's handwriting or one's height.

Q6 [4 pts]: What are advantages and disadvantages of using biometrics?

Some advantages to biometrics is that they are inherent within the entity, and therefore easy to always have on hand, are generally unique to the entity, do not need to be generated as they already exist, are hard to forge, and are generally universal.

Some disadvantages are that biometrics often contain more sensitive data than a password, etc., can be more detrimental if an attacker steals them, and difficult (or impossible) to change if they are compromised.

Authentication Protocols



Q7 [6 pts]: Figure above shows a challenge-response protocol for static biometric authentication. K_C is the shared key between the Host and the Client. B' is user biometric captured by device.

BT’ is the biometric template computed from B’. D’ is device authenticator computed by device. BT and D are biometric template and device authentication information at the Host. Match(BT' BT) returns 'yes' if the user computed biometric matches with stored biometric template at the host to within a certain pre-set threshold, and returns 'no' otherwise. Verify (D', D) check the validity of the authenticator and returns 'yes' or 'no'. If all verifications succeed at the host then the host returns 'yes' to client to indicate successful authentication.

i. **[3 pts] What purpose does random number r serve? Put another way, if the protocol is modified to not include r what vulnerability does this introduce?**

   The random number r serves as a freshness factor to verify that the response sent back by the user (which includes the biometrics and device data) is actually from this session (as the random number was produced in the previous step by the system and presumably has not been used before). Without this random number, a malicious user could send biometrics and/or device data produced at a different time, such as from a previous session they eavesdropped on. The random number links this set of data to this particular session.

ii. **[3 pts] Does message 3 from Client to Host need to be encrypted? Explain why. Specifically, won't integrity protection of this message using a keyed MAC be sufficient?**

   Message 3 does need to be encrypted, as if it were not, the biometric template and device data would be plain to see. An attacker could read and store this data for use at a later time and use it to impersonate the valid client. At a later date, they could open communications with the system to get the random number, and then just send it and the stolen data back, and the system would think it is the client. By encrypting with the client’s key, it protects this data and ensures that the client is the one which sent the message (as it is encrypted with their key).

   Using a MAC is not sufficient, as if the attacker steals the MAC (which is easier than the key used for encryption), they can impersonate the client by just sending that along with the data.

One-time passwords

Q10 [6 pts] Consider the hash function h(i) = (i + 5) mod 7, and suppose it is used in an implementation of the S/Key protocol. Let the seed be value 0, and suppose that the first password the user returns after the initialization step is 4.

(Passwords, starting with 0, are {0, 5, 3, 1, 6, 4})

i. **[4 pts] What password does the user return on the third login counting the first login password as 4.**

   The order of passwords is the reverse of the order they were calculated in, so if “4” is the first password, the user would use password “1” for the third login.

ii. **[2 pts] On receiving this password, the server (chose one action below)**

   (a) Is correct, as the server calculates the hash of the password you send (which would give password n+1, which is the previous password you sent).

   a. computes the hash of the returned password and admits the user if the hashed value is equal to the last correct password returned by that user

   b. computes the hash of the last correct password returned by that user, and admits him if that value is equal to the password just returned

   c. uses the initial key to recompute the 3rd password by repeated hashing, and admits the user if the recomputed value is equal to the password the user returns