University of Essex

Location Aware Password Protection System

School of Computer Science and Electronic Engineering

Name: Chathura M. Sarathchandra Magurawalage Supervisor: Prof. Kun Yang

Introduction.

LAPPS introduces extra layers of protection to the modern password protection systems. The implemented methodology can be applied to an array of scenarios. But for the sake of the implementation, using the same design, I concentrated on finding solutions to the vulnerabilities of debit/credit card pin/passwords that has been used on ATM machines. The emphasis is on warranting the following qualities of the passwords on generation.

- Location Awareness

The password is generated according to the user's geographical location. Hence the password is only active within a limited geographical area.

- Time Awareness

The password will only be authoritative for a limited amount of time (e.g. 5 minutes). After the time limit has been passed the password will be extinct.

- Dynamic

The password will be generated dynamically on demand.

- User Oriented/Specific.

A user only will have one password at a time. Password can only be used once.

Threats & Vulnerabilities in traditional ATM systems.

- Gaining confidential information by using skimming devices and hidden cameras.
- A potential criminal could be looking over of user's shoulder while typing the password.
- One password could be used on any ATM machine.
- Identity theft.

How LAPPS architecture harden the vulnerabilities in the traditional ATM systems?

- A password is allocated to the nearest ATM to the user, according to the user's geographical location. i.e The generated passwords can by used only on the allocated ATM machine. If someone got their hands on the generated password, they will have to use the password on the specific ATM.
- The password is generated dynamically, thus every time user logs in to an ATM, he/she will have a different password.
- The password can only be used once. Stolen passwords are useless.
- A user can only have one password at a time. This ensures that there are no any other active passwords for the same user.
- The password expires after 5 minutes.
- LAPPS 2-way Verification: To request a password using LAPPS, it is required that the user generates an 8 digit pin number from the PinSentry, using his/her pre allocated password and debit/credit card. This pin number is only valid for 1 minute.

How does LAPPS work?

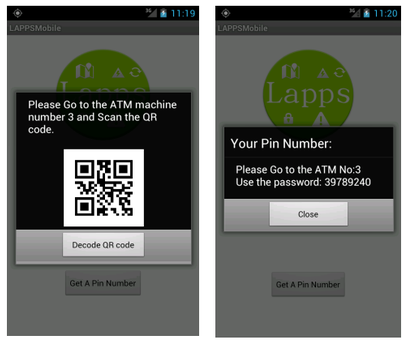
1. User Generates an 8 digit number using the pin sentry, by inserting their credit/debit card and the fixed password.
2. The User taps on the “Get Pin Number” button on the LAPPSMobile application, on the user's smart phone.
3. Insert the 8 digit number that has been generated using the PinSentry.
4. The mobile application sends the following information to the LAPPSServer.
   - The inserted 8 digit pin.
   - Users Geographical Information
   - Users unique ID
   - Registration ID of the application
5. If all above information are correct, then a password will be generated. The generated password and the user will be allocated to the nearest ATM machine.
6. The password and the allocated ATM machine number will be sent back to the user, encoded in a QR code. Connection is protected using TLS.
7. The user can either scan the QR code on the ATM or choose to type in the password.

PinSentry

How the server finds the nearest ATM?

- The server has got the latitude and longitude information of ATMs stored in a database, converted in to SRID 2770 format.
- After receiving the latitude and longitude information of the user, it converts them to SRID 2770 format.
- Then the server queries the database for the nearest ATM machines to the user, within 20m from the user.
- It orders the result by the distance, and gets the nearest one.

Screen Shots of LAPPSMobile.



The formula that has been used to create the 8 digit number.(PinSentry)

Hash() = SHA2-512

- Hash the user's fixed password
hPass = hash(static_passwd)

- Add the Time stamp rounded to the nearest minute. Then add the ID of the user at the end of the “hPass”.
tmpH=hash(hPass+timeStamp+userId)

- Convert tmpH to Hexadecimal.
- Retrieve the first four digits from the the hex string.
- Reverse the tmpH hex string.
- Retrieve the first four digits from the reversed string.