Password Storage and Retrieval

Nitin K N[1] Sharath S Chellappa[2] Chetan Purohit[3]
Madhusudan Thyagarajan[4]

[1]USN-1PI13EC064 Electronics and Communication Department
PES Institute of Technology

[2]USN-1PI13EC092 Electronics and Communication Department
PES Institute of Technology

[3]USN-1PI13EC025 Electronics and Communication Department
PES Institute of Technology

[4]Professor Electronics and Communication Department
PES Institute of Technology

Phase 3, 8th April 2017

Outline

Problem Statement

To design a bank system which ensures that compromise of any one server storing the password, never gives away the entire password of the user.

Timeline

- Feasibility Report - Literature survey, architecture of System.
- Phase 1 - Communication between bank and split servers, Basic layout of the app, partial implementation of splitting of password.[2]
- Phase 2 - Communication between app and bank server, development of login page of the app, issue of keys and securing communication along with hashing.
- Phase 3 - Optimisation and end case testing of communication, development of sign up page, generation of location pass and making app device specific.
- Phase 4 - Integration of application with the servers.
- Final Presentation - Validation of results by simulation.

Milestones

Inter Machine Communication

- Communication between app and bank server
  - Completion of the Signup and Login Portions of the communication diagram.
  - Serialization and transmission of packets to JSON objects.
  - Bringing about a link between independent platforms to ensure secure establishment of connection.
- Communication between bank server and split server
  - De-serialization of the JSON objects and extraction of data to tuples which can be referenced by the key.
  - Sending each of the corresponding parts of the split password to the split servers for storage and query.
  - Storage and retrieval from CSV files.

Architecture

Password Storage

User is asked to enter his registered bank account number and name.

$E_{skb}\{E_{pku}\{$Login id + bank account number + name$\}\}$

OTP + Security Question

OTP + Security Question

The Bank Server checks the answer and now splits the password and stores the different pieces in different servers with the hashed Login id being the query point. Location pass is generated conveys location of servers.

OTP + Answer + Password

$E_{skb}\{E_{pku}\{$OTP + User Password + login id$\}\}$

User Password Piece + Hashed(login id)

The user gets an acknowledgement that his password has been accepted.

Location pass to be hard coded into the app

Architecture

Password Retrieval

User clicks on the button to request access to his account

$E_{skb}\{E_{pku}\{\text{Login id}\}\}$

OTP

OTP

The Bank server correlates OTP with the login id. This correlation expires in a finite time.

The Bank Server decrypts the packet to get the login id and OTP, gets the location of the servers having pieces from the Location Pass and sends the login Id to each of the servers after checking if the OTP matches.

OTP + Password

$E_{skb}\{E_{pku}\{\text{OTP+login ID + Hash(User Password+OTP) + location pass}\}\}$

Hash(Login id) to each server

The user gets to access his account over a secure connection.

Returns Port number to set up connection with if there is a match

Each piece is sent to Bank Server

Milestones

Security aspects between the client and the bank server

- Mac address is used as the Unique identifier to ensure the device specificity of the client app. Hiccups were faced due to android 6.0 regulations.
- Implementation of 2 factor authentication using an OTP. The OTP is displayed as a distorted bitmap image which also authenticates that user is not a robot. (Implemented for both Login and Signup events.)
- Symmetric key encryption for all packets to be sent from the app to the server.
- App side validation of user input. Display of error messages in case of error.

Milestones

Problem encountered

When the bank server is compromised, the strength of the cryptosystem was no longer the entropy(guessing) of password but guessing of the servers containing pieces of password.

$$\text{Guessing of } p \text{ digit pin} = 10^p \quad (1)$$

$$\text{Guessing } k \text{ out of } n \text{ servers} = {}^nC_k \quad (2)$$

In our case,

$$\text{Guessing of 4 digit pin} = 10^4 \quad (3)$$

$$\text{Guessing 2 out of 4 servers} = {}^4C_2 \quad (4)$$

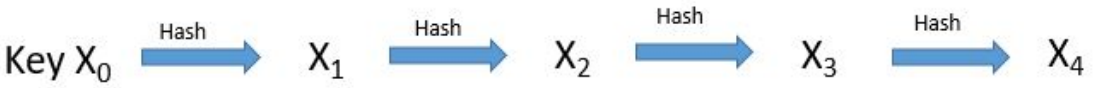
$${}^4C_2 < 10^4 \quad (5)$$

Strength of a cryptosystem should always be the entropy of the password, hence this problem needed to be looked into.

Milestones

Architecture of location pass and its implications

- Concept of hash chain being exploited to validate user(device specificity).



$X_1 = \text{Hash}(X_0)$ $\quad$ $X_3 = \text{Hash}^3(X_0)$

$X_2 = \text{Hash}^2(X_0)$ $\quad$ $X_n = \text{Hash}^n(X_0)$

- To reveal the location pass, involvement of 2 stakeholders being the bank server and the mobile device are needed.
- Adversary breaking into bank alone will need to guess location and one time key.

Architecture of location pass

Password Storage

Bank server

Split/subsidiary server

OTP + Login Id + hash(password+OTP)+ One time key (X_n)

Piece of password + One time key(Xn)

Bank server database

Name	Login Id	....	Enc(loc_id)X_n

Split server database

Hash(Login Id)	Enc(piece of pass)X_n

Architecture of location pass

Password Retrieval

Bank server

Split/subsidiary server

OTP + Login Id + hash(password+OTP)+ One time key (X_{n-1})

$Hash(X_{n-1}) \rightarrow X_n$

$Dec(enc(loc\_id)Xn)X_n \rightarrow loc\_id$

Hash(login_id)+ One time key X_{n-1}

$Hash(X_{n-1}) \rightarrow X_n$

$Dec(enc(piece\ of\ pass)X_n)X_n$

Piece of Password

Piece of Pass

$Enc(loc\_id)X_{n-}$

$Enc(piece\ of\ pass)X_{n-1}$