JOURNAL FOR INNOVATIVE DEVELOPMENT IN PHARMACEUTICAL AND TECHNICAL SCIENCE

Volume-2,Issue-9 (Sep-2019)
ISSN (O) :- 2581-6934

PAYMENT ASSISTANT

Neethu Prabhakaran P[1], Shejina N M[2], Safna K M[3], Dr.Carmel Prabha A[4]

[1, 2, 3,4]Assistant Professor, Department of Computer Science and Engineering, IES College of Engineering, Thrissur, Kerala

ABSTRACT:

Payment Assistant will provide fast and secure services to its users related to any type of payment and recharge. It enhances the security by providing a Web Client registration to get accesstotheapplication.AnumberoftechnologiesandstandardssuchasOpenIDandOauth have emerged to deliver an Internet-scale identity system during the past few years. Their evaluation results have demonstrated the difficulty of replacing passwords and highlighted the research challenges towards de-signing a password-less login scheme. Transactions can be done faster by storing details in the server database. Hence we are concluding that Payment Assistant App will provide a revolutionary step to the payment mechanisms.

INTRODUCTION

The easiest and cheapest way of authenticating an end user, password based authentication methods have been consistently chosen by almost every new cloud service. The explosive growth of cloud services and web applications has made it impossible for users to manage dozens of passwords for accessing different cloud services. Our application provides an easy way of accessing the payment method in a secured and easy manner. Payment assistant aims to improve on passwords with respect to both usability and security. This app is resistant to most common attacks on cloud services. The application of the proposed payment assistant security framework to the recent Mint Chip Challenge demonstrates the power of payment assistant for building a real-world password- less mobile payment solution.

Objective

A naive way to reduce users' burden for holding multiple passwords for different cloud services is to store users' credentials in a single device or service, and use certain key derivation functions to generate temporal passwords for sequential logins. However, this approach exposes the authentication server as a primary target of attackers.

SYSTEM DESIGN

Proposed system

Away to reduce users burden for holding multiple passwords for different cloud services is to store users credentials in a single device or service, and use certain key derivation Functions to generate temporal passwords for sequential logins. However, this approach exposes the authentication server as a primary target of attackers. The other approach is to employ an Internet-scale identity between web applications and cloud services. A number of technologies and standards such as Open ID and Oauth have emerged to deliver an Internet-scale identity system during the past few years. Their evaluation results have demonstrated the difficulty of replacing passwords and highlighted the research challenges towards de-signing a password-less login scheme. In this contribution, payment assistant, an innovative security frame work for password-less universal login. After an initial registration process, Payment assistant enables a user to access multiple cloud services or web applications with only few taps on his/her mobile devices. This salient feature comes from the adoption of push message services for mobile devices and public-key cryptography. Different from most existing login solutions, the servers in payment assistant are not able to generate users' credentials. Therefore, even if a Payment server is compromised, attackers cannot impersonate a user in order to access cloud services. As a potential application of the payment assistant security framework, we have applied it to build a password-less mobile payment solution for tackling the recent Mint Chip Challenge system that defines standardized mechanisms enabling the identity attributes of its users to be shared.

Advantages

- Loxin servers helps in not to access user credentials

- Attackers cannot impersonate to access cloud services
- Solution for password less mobile payment process
- Users satisfaction and security

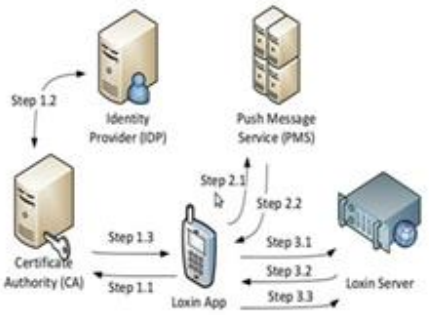

Figure 1 System Architecture

Step 1 Obtain a public-key certificate from CA.

Step 1.1 The Loxin App generates a pair of public key PK and private key SK. The Loxin App prompts the user to choose or enter an ID (e.g., email address) and then sends ID and PK to the CA.

Step 1.2 The CA first communicates with the IDP and verifies the user's ID, such as sending a verification email to the claimed address.

Step 1.3 if the user's ID is verified, the CA sends it's signed certificate Cert (ID, PK), containing both ID and PK, back to the Loxin App.

Step 1 is only required to be completed once. After that, the user can log in to other cloud services by using this ID. Please note that the private key SK should be securely stored, and never be released outside the Loxin App.

Step 2 Register to a PMS.

Step 2.1 The Loxin App sends a registration request to a PMS.

Step 2.2 The PMS verifies the request and sends back credentials for registration, which can be used by other software and services to send messages to the Loxin App. Here we simply use a token Tok to represent all the PMS credentials.

Step 3 Register to the Loxin Server securely.

Step 3.1 The Loxin App sends a registration request, which contains Cert (PK, ID) and Tok, to the Loxin Server.

Step 3.2 The Loxin Server responses with a random number Rreg and an expiration time Treg for this request.

Step 3.3 The Loxin App signs ID, Tok, Rreg and Treg with its private key SK. The signature

Module description

Payment assistant contains mainly 5 modules they are,

- Loxin client
- Certificate authority
- PMS server
- Loxin server
- Web client

Loxin client

A loxin app is created to act as a loxin client. Once the user installs the loxin app it performs a one-time registration process. The loxin app generates a pair of public key PK and private key SK. The loxin app prompts the user to choose or enter an ID and then sends ID and PK to CA.

Certificate authority

The CA first communicates with the IDP and verifies the user's ID such as sending and verification email to the claimed address. If the user's ID is verified the CA sends it's signed certificate Cert (ID, PK), containing both ID and PK, back to the Loxin APP.

PMS server

The Loxin APP sends a registration request to a PMS. The PMS verified the request and sends back credentials for registration, which can be used by other software and services to send messages to the Loxin App. Here a simple token TOK to represent all PMS credentials

Loxin server

The Loxin App sends a registration request which contain Cert (PK, ID and Tok, to the Loxin Server. The Loxin server responses with a random number Preg and an expiration time T for this request. The Loxin App signs ID, Tok, Rreg, and T with its private key SK. This signature is sent to and verified by the Loxin Server. If the signature is valid the Loxin Server stores the pair (ID, Tok) into its database for later use.

Web client

The web client here acts as a website that request authentication of a particular user so as to grant access to its services to the user.

ISSN:-2581-6934|www.jidps.com

SYSTEM IMPLEMENTATION

Authentication

Users can authenticate their pre-owned identities to various cloud services even without pairing with or registering to those services first. This feature is able to remove or shorten registration processes and make cloud service more user-friendly. When a user wants to log into a cloud service from his/her computer a back end server of the cloud service will generate a random challenge for the user, and the Server will forward the challenge to the App via the PMS. Upon receiving the user's manual permission, the App will sign the challenge with the private key SK and send the signature to the cloud service for verification.

Step 1: The user enters and submits only ID to the cloud service.

Step 2: The cloud service generates a random number Rauth, an expiration time Tauth, and a callback address URL for this login request. In addition, a cryptographic hash value tag = hash (ID, Rauth, Tauth, URL) is computed and displayed on the user's computer. The hash value may be rep- resented by certain formats, such as figures or colorful barcodes, other than plain strings, so it can easily be visually checked by the user.

Step 3: The cloud service sends ID, Rauth, Tauth, and URL to the Server.

Step 4: The Server searches ID in its database in order to retrieve the corresponding Token The Server then uses Tok to send Rauth, Tauth, and URL to the PMS. Step 5: The PMS forwards Rauth, Tauth, and URL to the user's App.

Revocation

When a user's phone is lost, the private key SK stored in the App might be compromised either. In this case, the user needs to contact the CA to revoke the certificate of the corresponding public key PK. For example, if the CA allows only one certificate for each ID, the user may go through the registration process again to revoke the old certificate. Contacting the CA to revoke the lost certificate may be time-consuming, and the user's email account may be compromised as well once the mobile device is lost. One possible solution to secure the revocation process is generating a second pair of public and private keys, PK and SK, during the registration process. This second key pair should be stored out of the mobile device, e.g. printing on a paper, for security purpose. If the user's primary secret key SK may be leaked, the user can authenticate his/her identity by using PK and SK to the Loxin server to block further login requests associated with the leaked PK. PK and SK may also be used by the CA to verify users in order to revoke certificates. In order to minimize the risk that the user's private key is used by adversaries, certain counter measures may be deployed, e.g., requiring a short PIN to access the App and limiting the number of retrials. Please note that adding such a PIN will make the application less convenient, but it is still much more user-friendly than remembering and entering long passwords. Moreover, other information such as fingerprints and network locations may be considered to unlock the application instead of short PINs in the future, in order for improving usability and security.

RESULTS

Figure 2 User registration

Figure 3 Login after registration

ISSN:-2581-6934|www.jidps.com

Figure 4 Home screen of application

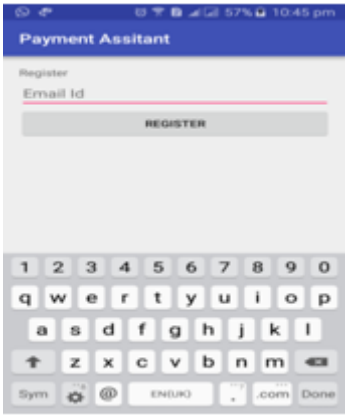

Figure 5 Login page

Figure 6 one time Verification Code

Figure 7 Entering Code for login

Figure 8 Bank Details

CONCLUSION

Android payment applications with their features and compared with the newly developed android app "PAYMENT ASSISTANT" and concluded that this application will provide fast and secure services to its users related to any type of payment and recharge. It enhances the security by providing a Web Client registration to get access to the application. A number of technologies and standards such as Open ID and Oauth have emerged to deliver an Internet-scale identity system during the past few years. Their evaluation results have demonstrated the difficulty of replacing passwords and highlighted the research challenges towards de-signing a password-less login scheme. Transactions can be done faster by storing details in the server database. Hence we are concluding that Payment Assistant App will provide a revolutionary step to the payment mechanisms.

Future work

Current System save the login details in the database and multi users cannot access so by providing more login spaces that can be done. More UI design can be added for the better appearance

REFERENCE

[1] (2012, Feb.). The Science Behind Pass faces [Online]. Available: http: //www.Realuser.com/published/ScienceBehindPassfaces .pdf

[2] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, andA. Rubin, "The design and analysis of graphical

ISSN:-2581-6934|www.jidps.com

passwords," in Proc. 8th USENIX SecuritySymp.1999, pp.1–15

[3] H. Tao and C. Adams, "Pass-Go: A proposal to improve the usability of graphical passwords," Int. J.Netw. Security, vol. 7, no. 2, pp.273–292, 2008.

[4] S. Wieden beck, J. Waters, J. C. Birget, A.Brodskiy, and N. Memon, Pass Points: Design andlongitudinal evaluation of a graphical passwordsystem," Int. J. HCI, vol. 63, pp. 102– 127, Jul. 2005[5] P. C. van Oorschot and J. Thorpe, "On predictive models and user drawn graphical passwords," ACMTrans. Inf. Syst. Security, vol. 10,no. 4, pp.1–33,