Visual Authentication – A Secure Single Step Authentication for User Authorization

Luis Roalter [1], Matthias Kranz [2], Andreas Möller [1], Stefan Diewald [1],
Tobias Stockinger [2], Marion Koelle [2], Patrick Lindemann [2]
[1] Technische Universität München, Munich, Germany
[2] Universität Passau, Passau, Germany
roalter@tum.de, matthias.kranz@uni-passau.de, andreas.moeller@tum.de,
stefan.diewald@tum.de, tobias.stockinger@uni-passau.de,
marion.koelle@uni.passau.de, patrick.lindemann@uni-passau.de

ABSTRACT

User authentication on publicly exposed terminals with established mechanisms, such as typing the credentials on a virtual keyboard, can be insecure e.g. due to shoulder surfing or due to a hacked terminal. In addition, username and password entry can be time-consuming and thus improvable with relation to usability. As security and comfort are often competing with each other, novel authentication and authorization methods especially for public terminals are desirable. In this paper, we present an approach on a distributed authentication and authorization system, where the user can be easily identified and enabled to use a service with his smartphone. The smartphone (as personal and private device the user is always in control of) can provide a highly secure authentication token that is renewed and exchanged in the background without the user's participation. The claimed improvements were supported by a user survey with an implementation of a digital room management system as an example for a public display. The proposed authentication procedure would increase security and yet enable fast authentication within publicly exposed terminals.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; K.6.5 [**Security and Protection**]: Authentication

Keywords

Authentication, Authorization, Public Displays, Intelligent Environment, Security, Smartphones

1. INTRODUCTION

Authentication and authorization, while mandatory and necessary, are often perceived as a burden by the user. The acceptance of lengthy authentication procedures is often low when users need to login on a potentially insecure and untrusted terminal, e.g., public displays or foreign computers. For such situations, a simple and secure solution based on single-sign-on can be offered. This login ticket, including a so-called token, would be generated together with the remote identity provider (IdP) and transmitted to the user's smartphone once. As this ticket is only valid for a limited period of time, hacking attempts would be hindered, as the user's credentials are no longer visible to nearby persons. This approach makes user authentication less error-prone and faster, as authentication on the smartphone is done in a *single step* (scan token and the smartphone automatically does the authentication) over an auxiliary channel (e.g. auditive, visual, NFC etc.).

In this paper, we present the architecture and evaluation of a usable and secure authentication system that uses the smartphone as credential provider. The smartphone can automatically generate an authentication token, which is used to authenticate the user on various networked devices.

In the upcoming section, we discriminate our approach from related work. Subsequently, we present our proposed system in a common scenario. We conclude with an outlook on future work.

2. RELATED WORK

A look at recent trends in research and industry shows that multi-factor authentication is becoming more popular. In standard authentication and authorization procedures, the user provides login name and password to a known login mechanism. To enhance security, numerous companies (e.g. Amazon, Google, Facebook, Microsoft, PayPal or Twitter) are securing their web services with multi-factor authentication (MFA) models [1] to avoid manipulation in security-critical processes (e.g. changing personal details, passwords, phone numbers, starting money transfer, etc.). Banking institutions use additional secrets to secure each transaction with so-called transaction authentication numbers (TANs). TANs are similar to one-time passwords, which are calculated on-demand and are only valid for a limited period of time [2]. These approaches can already make use of additional devices, independently from the current terminal [3, 4]. Additionally, several approaches have been presented for increasing security and reducing the risk of shoulder surfing when entering personal information on a smartphone, e.g. gesture-based authentication [5, 6].

One way to increase security is by combining different *factors* to an authentication token (like requiring both a physical key and a number to unlock a safe). Current approaches contain the following three factors:

MUM '13, Luleå, Sweden

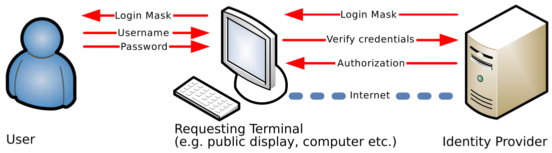

Figure 1: After receiving the login mask from the server (e.g. a form containing username and password fields), the user enters his credentials. These credentials are validated by the IdP and the user will be authenticated and authorized by the server.

1. The login name, username or email address for determining the user account
2. The user's password
3. A potential periodically changing secret, provided by an external or independent device.

The third factor (e.g. RSA SecurID [7]) or a dedicated smartphone app (e.g. Google Authenticator[1]) can be omitted when the requesting device (e.g. by IP and MAC address) is already known to the system (e.g. with web services from Google, Facebook or Twitter). The omitted factor will then only be requested when dealing with security-critical functions (e.g. money transfer, password changes). This additional authentication factor increases security, but not comfort of the authentication procedure, as an additional action is required from the user. For authentication and authorization, the following models are used:

Centralized Authentication: The secrets of the user (both password and token challenge) are transmitted securely to and verified on a central administrated server (e.g. Kerberos, LDAP). The advantage of this approach is the easy management of the user's secrets and data. The computer on which the user wants to authenticate needs to establish a connection to the IdP and verify the credentials. This combined authentication and authorization process, as shown in Figure 1, is the prevalently used authentication mechanism nowadays. However, the approach shows weaknesses when authenticating on a publicly accessible terminal. The terminal does not (except for some smartcards) present its identity for verification to the user. Moreover, public terminals could have been compromised, or their traffic could be redirected over a hijacked server.

(Partially) Local Authentication: At least one secret can be locally calculated and verified, using a shared or preshared key of the user and provider (e.g. time and counter-based one-time passwords). The advantage of one-time passwords (OTP) is that the secret pass-phrase to generate the OTP never has to be transmitted over a (potentially insecure) connection [2]. In contrast to centralized authentication, the terminal or smartphone can already verify a provided secret without transmitting any data over a probably insecure channel [8], e.g. by calculating challenges from a known secret and current time. However, the calculated hash value and user credentials still need to be transmitted to a central authentication and authorization service, as shown in Figure 2. In contrast to plain username and password authentication mechanisms, the token is usually only valid for a short time and will change the next time a token is requested. This complicates a man-in-the-middle attack, as simple replay attacks of the user's credentials are not successful [9, 10].

Session-Based Authentication: The session is transmitted from the terminal to the user's smartphone and verified independently from the terminal. The actual authentication on the smartphone can be done using conventional mechanisms. The difference in this approach is that although the token for the session is still generated on a probably insecure terminal, no user-related data is transferred to the terminal during the login process. Session-based authentication bases on the the use of auxiliary channels [11, 12].

3. SYSTEM ARCHITECTURE

We use the authentication architecture presented by Roalter et al. [13]. In contrast to previously discussed approaches, we eliminate the usage of the keyboard during the authentication procedure, omitting the user interaction step of typing personal credentials. The user only needs to authenticate once against his IdP on his personal smartphone. However, this authentication process is independent from the authorization process and can therefore be done at a private device. This increases security and privacy during the authentication when using a public terminal.

As a consequence, the user is no longer known to the system by his user name until the IdP authorizes the usage of the request resource. Decoupling the authentication from the authorization procedure and actual service entails some challenges for both security and privacy.

3.1 Session-Based Authentication

Session-based authentication includes the problem that the user name is not known in advance. In our solution, first of all a unique session ID (SID) in form of a 256-bit token out of a SHA-1 hash function, is generated for the terminal by the IdP. This SID is anonymous and valueless until the session gets authenticated. Moreover, the SID is only valid for a short time. Once the session expires, the terminal needs to renew the SID (see Figure 3). The user can use the SID presented by the terminal (e.g. visualized as a QR code) and insert the SID in his authentication program. Even if someone else uses the camera to scan the QR code, no interference is possible during the authentication process as no user-related data is encoded in the visual code.

3.2 Remote Administration

As the session management has been moved to the IdP, the user can obtain an overview on his currently authenticated sessions from every terminal (both for online sessions like web services and local sessions like workstation logins). The remote administration is an interface to the IdP, receiv-

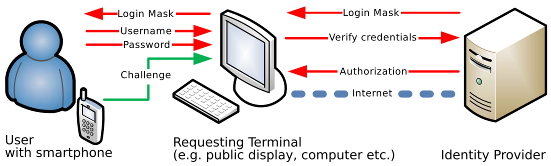

Figure 2: In this example, the user needs to type in an additional challenge from an external device (e.g. personal smartphone) in conjunction with username and password. The code can be calculated by a known hash function or is transmitted via a push notification service to the user's device.

[1] Google Authenticator, *http://code.google.com/p/google-authenticator*

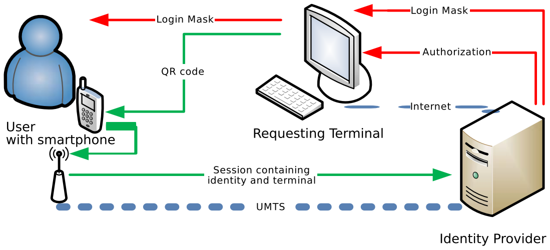

Figure 3: Authentication is handled by the user's smartphone. The smartphone scans a session-related code (e.g. token) and can (automatically) authenticate the user by connecting to the identity provider (IdP) on an independent auxiliary channel (e.g. over UMTS/EDGE).

ing recent information about the user's sessions. As a unique session ID is stored on the server, running sessions can also be transferred to a new terminal in the same infrastructure.

4. IMPLEMENTATION AND EVALUATION

We evaluated the system architecture at the example of an interactive public display, used for room management. The system allows, e.g., authorized room access. We consider this as suitable scenario for quick, yet secure one-step authentication without the need to enter personal data on the public screen. The process is depicted in Figure 4.

4.1 Authentication Process

The one-step authentication requires an initial, conventional authentication procedure on the user's smartphone, using a native Android app we implemented. The smartphone authenticates against the IdP and receives a personalized token (used ID). Username and password are not stored on the smartphone.

Once the token is saved within the application, the user can use it to authenticate with a web service (e.g., on a public display) with username and password (and optional additional credentials). The web service provides a login token transportation mechanism, which allows the user to scan a QR code from the public screen integrating the SID. Authentication token and SID are transmitted to the IdP, where the user is authenticated to the requested service.

Alternatively, the user can authenticate with username and password in a standard login mask. Doing this on a touch-based device can be challenging, especially when the password contains alphanumeric or special characters. However, this authentication procedure is straightforward and well-known, and can serve as fall-back solution for users not owning a smartphone.

The personalized authentication token on the smartphone needs to be renewed periodically. A system with high safety standards might request a more frequent renewal of the token. Further, this limited lifetime of the token allows the user to revoke the token of (previously) authenticated sessions remotely with his smartphone or via the web front-end (e.g., when the smartphone gets lost, a session is still running etc.).

4.2 Large-Scale Applicability

The proposed one-step authentication procedure is applicable for multiple sessions on different devices. It can thus

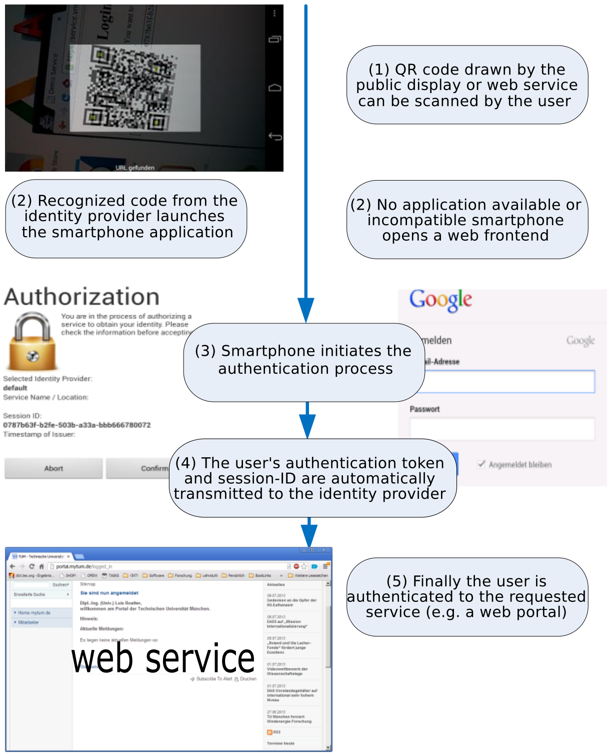
(1) QR code drawn by the public display or web service can be scanned by the user
(2) Recognized code from the identity provider launches the smartphone application
(2) No application available or incompatible smartphone opens a web frontend
Authorization
Google
(3) Smartphone initiates the authentication process
Abort
Confirm
(4) The user's authentication token and session-ID are automatically transmitted to the identity provider
web service
(5) Finally the user is authenticated to the requested service (e.g. a web portal)

Figure 4: The user scans (1) the QR code, which is interpreted in (2) by the smartphone by starting the appropriate application (left) or opening a web page (right). In (3) the user provides his credentials either by confirming the authentication, or by authenticating with username and password. In the background (4) the credentials are validated on the IdP. Both ways result in accessing (5) the requested (web) service.

also used in large scale on service or application level (e.g., corporate email programs). Therefore, their authentication procedure must be replaced by an additional system. As the single-step authentication hides the user name in the beginning, existing systems as GINA[2] (Graphical Identification and Authentication), Credentials Provider[3], or the Open Pluggable Authentication Module (OpenPAM)[4] do not support single-step authentication out of the box.

We prototypically implemented a PAM module for Linux which is able to visualize the QR code and request additional data from the IdP in the background. When the user scans the QR code, the user's token is authorized to use a service by transferring the session ID in combination with the user's token to the IdP. As the user does not input his username anymore, his identity cannot be exposed to others, as the handshake is done in the background between smartphone, IdP and the terminal. In the internal database of the IdP, both session ID and user token already exist. As the IdP knows the device the session was registered from, it can grant access to the resource by simply pushing a notification to the service. This authorization grant includes

[2] Graphical Identification and Authentication (GINA) *http://msdn.microsoft.com/en-us/library/aa380543.aspx*

[3] Credentials Provider, *http://msdn.microsoft.com/en-us/library/windows/desktop/bb648647.aspx*

[4] OpenPAM, *http://openpam.org*

the user's name and additional user-related information the service is allowed to receive. The same procedure is used to authenticate the user against any web service used on an authorized computer.

4.3 Session Management

For each user, the IdP keeps a list of the authorized sessions and the devices the sessions were authorized with. This central session management allows the user keep track of all his authorized sessions. As SIDs need to be regularly renewed, the user can invalidate any session from his smartphone or any other computer connected to the Internet. The user can revoke all currently authorized sessions which are grouped by the authenticating device (e.g. authenticated by smartphone A, smartphone B or authenticated by a conventional login form). This allows both the user and the companies the user is working with to increase accountability, as they are able to verify if a stranger is using a foreign session. This verification can be automated, as the smartphone application is notified when a new session was authorized.

4.4 User Survey Results

The prototypical implementation was evaluated on a public display. In this context, we picked two important research questions from the user study to learn

- *whether users would have concerns about entering personal credentials on a public display (RQ1)* and
- *whether users would have privacy concerns when using our system (RQ2).*

The study was conducted with 20 subjects (18 males, 2 females), aged between 20 and 64 years. Subjects agreed with the statement that they have security concerns when entering credentials on public terminals (RQ1) with averagely 3.8, standard deviation (SD) = 1.3, on a 5-step Likert scale (1 = fully disagree, 5 = fully agree). By contrast, they agreed significantly less with the statement that our smartphone-based authentication approach would entail privacy concerns, addressing RQ2 (avg. 2.3, SD = 1.4).

5. CONCLUSION AND OUTLOOK

The presented approach shows that it is possible to increase security for authentication and authorization procedures. We use a native application running on an Android smartphone. A limitation of our approach is the requirement of additional (external) hardware (e.g. smartphones). In case of the unavailability of such a private device, credential-based authentication is still available as fallback. The fact that no credentials must be entered any more and the user survey results suggest that our implementation provides a more intuitive, easy and secure authentication and authorization for daily-used services.

In future work, the architecture will be improved by establishing a secure channel between smartphone and IdP. Smartphones can have problematic security leaks [14]. As an example, an installed malware application could redirect any authorization procedure over a custom proxy server. Such attack vectors must be analyzed prior to a larger distribution of this authentication procedure. For usage on standard personal computers, authentication and authorization procedures must be connected to that applications running in user space can benefit from one-time authentication as well. Finally, the complete system will be implemented and evaluated in a larger context. This allows to investigate how the system can cope with users' daily needs authentication and authorization needs, especially in enterprise environments. This includes, e.g., room management and entrance systems (e.g. public displays capable to unlock a room), login on the company network and computing system, and the integration into an existing single-sign-on provider as they are already used in larger companies.

6. REFERENCES

[1] Dena Terry Bauckman, Nigel Paul Johnson, and David Joseph Robertson. Multi-Factor Authentication, 2013. US Patent 20,130,055,368.

[2] Neil Haller, Craig Metz, Phil Nesser, and Mike Straw. A One-Time Password System. Technical report, 1996.

[3] Günther Starnberger, Lorenz Froihofer, and Karl M. Göschka. QR-TAN: Secure Mobile Transaction Authentication. In *Intl. Conf. on Availability, Reliability and Security*, pages 578–583. IEEE, 2009.

[4] Anna Vapen, David Byers, and Nahid Shahmehri. 2-ClickAuth Optical Challenge-Response Authentication. In *Intl. Conf. on Availability, Reliability, and Security*, pages 79–86. IEEE, 2010.

[5] Alexander De Luca, Martin Denzel, and Heinrich Hussmann. Look Into My Eyes! Can You Guess My Password? In *5th Symp. on Usable Privacy and Security*, pages 7:1–7:12. ACM, 2009.

[6] Shwetak N. Patel, Jeffrey S. Pierce, and Gregory D. Abowd. A Gesture-Based Authentication Scheme for Untrusted Public Terminals. In *17th Symp. on User Interface Software and Technology*, pages 157–160. ACM, 2004.

[7] Alex Biryukov, Joseph Lano, and Bart Preneel. Cryptanalysis of the Alleged SecurID Hash Function. In *Conf. on Selected Areas in Cryptography*, pages 130–144. Springer, 2004.

[8] Min Wu, Simson Garfinkel, and Rob Miller. Secure Web Authentication with Mobile Phones. In *DIMACS Workshop on Usable Privacy and Security Software*, pages 9–10, 2004.

[9] Bruce Schneier. Two-Factor Authentication: Too Little, Too Late. *Comm. ACM*, 48(4):136, 2005.

[10] Rolf Oppliger, Ralf Hauser, and David Basin. SSL/TLS Session-Aware User Authentication. *Computer Comm.*, 41(3):59–65, 2008.

[11] Yi-Pin Liao and Shuenn-Shyang Wang. A Secure Dynamic ID Based Remote User Authentication Scheme for Multi-Server Environment. *Computer Standards & Interfaces*, 31(1):24–29, 2009.

[12] Rene Mayrhofer, Jürgen Fuß, and Iulia Ion. UACAP: A Unified Auxiliary Channel Authentication Protocol. *Transact. on Mobile Computing*, 12(4):710–721, 2013.

[13] Luis Roalter, Matthias Kranz, Stefan Diewald, and Andreas Möller. The Smartphone as Mobile Authorization Proxy. In *14th Intl. Conf. on Computer Aided Systems Theory*, pages 306–307, 2013.

[14] Andreas Möller, Florian Michahelles, Stefan Diewald, Luis Roalter, and Matthias Kranz. Update Behavior in App Markets and Security Implications: A Case Study in Google Play. In *Proc. of the 3rd Intl. Workshop on Research in the Large . Held in Conjunction with Mobile HCI*, pages 3–6, 2012.