Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Design and Evaluation of an Architecture for Ubiquitous User Authentication based on Identity Management Systems

Marc Barisch
Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, 70569 Stuttgart, Germany
marc.barisch@ikr.uni-stuttgart.de

***Abstract*—Nowadays, users consume digital services with their digital identities on a multitude of different devices, e.g. notebooks, smartphones or even TV sets. Hereby, users are faced with additional challenges, i.e., devices have different security levels and not all digital identities must be used on all devices. Identities used for home banking should not be used on an insecure device and business identities should only be used on business devices. Moreover, it should be possible to switch between devices in a seamless way without the need to reauthenticate again on each device. Therefore, we propose an architecture that integrates all user devices and exploits identity management systems for ubiquitous user authentication.**

The proposed architecture improves usability by reducing the number of manual authentication procedures, by relaying authentication to devices with appropriate input capabilities and by supporting the user in identity selection. Security is improved by the possibility to perform authentication on secure devices, the provisioning of short-lived tokens to insecure devices and the opportunity to perform multifactor-authentication across devices. Our implementation is based on the Shibboleth IdM system and serves as proof-of-concept of our architecture. The conducted security evaluation confirms that our concept does not introduce additional security threats.

***Keywords*-Digitial Identity, Identity Management, Ubiquitos authentication, Virtual Device, Session Management**

I. Introduction

Since a couple of years we realize that Marc Weiser's vision of ubiquitous computing is becoming reality. Usage of information technology became an integral part of our daily life and makes it easier. However, we must admit that we still have not achieved the principle of disappearing information technology in the sense that we are not "freed to use them without thinking" [1].

One prominent example that confirms this statement is the authentication against services. Nowadays, we struggle with multiple accounts for various service providers (SP). Each requires us to memorize password-username combinations for authentication. This contradicts to the vision of disappearing information technology in two aspects: Usability and Security.

It is obvious that memorizing a multitude of username-password combinations is not usable. Therefore, users tend to use simple passwords that can be revealed by brute-force attacks. In addition, they use the same username-password combination with different SPs. This makes the user vulnerable to various attacks. Among them are impersonification attacks by malicious SPs, pishing attacks and attacks against the SP's user database that often store user passwords in cleartext as recently exploited [2], [3]. Moreover, usability is degraded with the introduction of different authentication methods. In particular for online-banking and business applications, the used authentication methods are more sophisticated and more challenging to users.

Some of the mentioned drawbacks are solved by user-centric identity management (IdM) systems like OpenId [4], Shibboleth [5], Liberty Alliance [6] or Microsoft Cardspace [7]. By means of single sign-on (SSO) and federation the number of username-password combinations can be reduced significantly. Instead of having a username-password combination for each SP individually, the user authenticates against an identity provider (IdP) that is trusted by SPs. In addition the IdP can provision user attributes (e.g. age, address) to SPs and increase the usability neglecting potential privacy drawbacks. In the following we call an account with an IdP a digital identity.

Most IdM systems implicitly assume that the user has only one device to use his digital identities in order to consume services. This assumption does not hold anymore. Today a user possesses and even simultaneously uses different devices (e.g. smartphone and notebook) for varying purposes (e.g. private, business). This creates not only additional challenges regarding the security and usability of IdM systems, but also provides new opportunities like multi-factor authentication. Among the challenges and opportunities are:

- Seamless device change: Consumption of services across devices without the need to reauthenticate for every service individually on each device.
- Sharing of security features: Make use of security capabilities (e.g. authentication methods) that are provided by another device that is owned by the same user for secure authentication.
- Authentication on secure devices: Use the most secure

user device for authentication.

The focus of this paper is on the presentation of an architecture that tackles the challenges and exploits new opportunities, which emerge from using several devices with IdM systems. This paper is structured as follows. In Section II, we introduce five usage scenarios that illustrate challenges and opportunities to extend IdM in general and in particular authentication to multiple devices. Based on the scenarios, we derive core requirements (c.f. Section III) that are addressed by three key concepts in Section IV. The key concepts are reflected in the overall architecture that is presented in Section V. Section VI describes the implementation based on the Shibboleth IdM system. In Section VII we show how we evaluated the security of the architecture. Section VIII presents related work before Section IX concludes the paper.

II. Usage Scenarios

Based on the assumption that a user has several devices, we introduce five different scenarios that identify new challenges for IdM systems.

A. Scenario 1: Fast Device Change

Mobility is becoming commodity. Many activities are performed at different locations with different devices. One visionary scenario, which is often used in research [8], [9], is the change of devices due to mobility.

Description: At home, the user begins watching a movie provided by a video on demand (VoD) provider on the 50" TV screen. That means the user has to authenticate against the VoD provider on the TV screen. During the movie, the user leaves his home and wants to continue watching on the smart phone. Today, the user has to reauthenticate, select the movie again and trigger a fast forward to the position where the movie was stopped.

Challenge: It should be possible to continue an existing service session on a second device without the need for reauthentication against the SP on the second device.

B. Scenario 2: Insufficient Security Features

Some services need more trust into the user identity than others. This can be achieved by the usage of dedicated security equipment (e.g. card readers or one-time password generators).

Description: The SP requests that authentication should be based on a SIM card, since the operator of the mobile network is trustworthy and has verified the identity of its customers by out-of-band means (e.g. verification of passport). Since the notebook of the user has no means for SIM card based authentication, the user cannot make use of the service.

Challenge: It should be possible to share the authentication capabilities across all devices of a user.

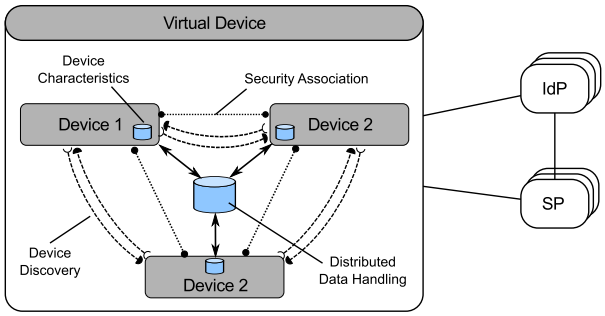


Figure 1: Virtual Device concept

C. Scenario 3: Business and Private Devices

Many employees use notebooks, smartphones and other devices that are provided by their employer. In addition, every employee has its own private devices. In many jobs the border between private life and business activities is blurred. That means one can work at home or use time on business trips for private purposes.

Description: A consequence of such nomadic behavior is that you often do not differentiate between private use and business use of your communication devices. For example one checks business mails on private computers via web interfaces or uses private Facebook accounts on the business smartphone. Hereby, every usage context has different security requirements [10].

Challenge: The usage context of identities and devices should be considered and identity usage potentially restricted.

D. Scenario 4: Identity Usage on Insecure Devices

Often users have computers or other communication devices at home that are not that good maintained from a security perspective. Either necessary security patches are not applied leading to a lot of vulnerabilities or malicious programs are installed. Therefore, using such kind of devices might have serious consequences. In particular authenticating on such devices might lead to intercepted credentials (e.g. username/password combinations) resulting in identity theft and impersonification.

Description: A user wants to read his emails provided by a webmail provider on an insecure machine. Since the machine has generous hardware (large display, ...) it is attractive for the user, even if the machine does not provide adequate security. The smartphone, which is assumed to be more secure, is not used at all.

Challenge: For authentication, i.e. the usage of credentials, the most secure device should be used. Only short-time credentials should be made available to insecure devices.

E. Scenario 5: Insufficient Input Methods

More and more devices get network access without sophisticated input capabilities, like keyboards. A user should be able to authenticate on a device with appropriate input capabilities.

Description: If a user wants to access his private images on a game console, it should be possible to use the notebook for authentication [11].

Challenge: Relay the authentication to a more powerful and trusted device in case of limited input capabilities.

III. Requirements

Based on the usage scenarios introduced above, we derive the following requirements:

- R1 - Secure exchange: We have to securely exchange information between the user devices. Among the information might be assertions for authentication against SPs.
- R2 - Task distribution: Different devices of a user are responsible for different tasks. It must be possible that authentication against an IdP takes place on one device, whereas the service is consumed on another device.
- R3 - Remote Activation: If one device cannot fulfill the requirements, e.g. by a SP regarding authentication, it must be possible to trigger another device to perform the authentication.
- R4 - Discovery of user devices: It is required to discover all devices belonging to the same user. Only devices in the proximity of the user can be used for authentication against IdPs.
- R5 - Capture of device characteristics: The properties of user devices have to be captured and exchanged among each other. Supported authentication methods are of interest in particular. In addition it is required to capture relevant device properties (e.g. operating system, installed software) to determine the security level of a device. The security level of a device is a metric that allows the quantification of security.
- R6 - Establishment of security associations: It is necessary that devices authenticate each other and establish a confidentiality and integrity protected channel.
- R7 - Determination of usage context: The usage context of devices as well as of identities has to be declared by the user.
- R8 - Distributed data handling: Every device captures and stores information regarding the device itself as well as the user and his identities. Since this data is required for decision making, it has to be exchanged between the devices.

These requirements are addressed by the following key concepts. The key concepts have been defined to separate concerns in the design of the architecture.

IV. Key Concepts

A. Virtual Device Concept

The Virtual Device is the basic key concept. It provides an umbrella for all devices belonging to one user and renders it possible that all user devices appear to 3rd parties (i.e. SPs and IdPs) as one device. Fig. 1 illustrates this concept.

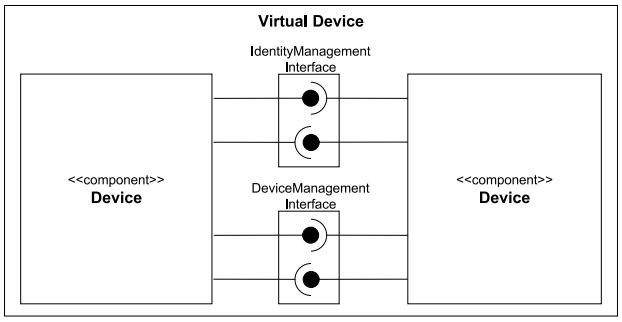


Figure 2: Virtual Device View

This concept is not new. In literature several proposals exist that have designed architectures to integrate various devices and make shared use of the available resources [12]–[14]. We use it as a basic enabler to fulfill the following requirements: R1, R4, R5, R6, and R8

B. Session Split Concept

Ordinary IdM systems are designed to have the IdP session and the SP session on the same device. The IdP session is established by authenticating against the IdP. Based on an IdP session, the user can obtain assertions and provide these assertions for authentication to SPs. The SP verifies the obtained assertion and grants access to its service. Depending on the IdM system, the SP and the IdP have established a trust relationship apriori (e.g. Shibboleth and Liberty Alliance) or establish it on demand (e.g. OpenId).

Based on the Virtual Device concept, we assume trust between devices. Therefore, we can basically distribute sessions across devices. The device having established the IdP session needs not necessarily to be the same device as the one on which the service is consumed. This enables a new degree of freedom, because new selection criteria for distributing sessions are available. For the selection of a device to establish the IdP session the following criteria can be considered:

- Security level
- Authentication capabilities
- Input devices
- Usage context

In contrast, the SP session might be established on devices that are more powerful or dedicated regarding resources (e.g. display size, computing power) like game consoles. With this key concept we address the requirement of task distribution (R2).

C. Multi-device IdM Concept

Based on the Virtual Device concept and the Session Split concept, we can establish the Multi-device IdM concept. The Multi-device IdM concept comprises the secure exchange of assertions to establish SP sessions, the remote activation of identities on other user devices, the prefiltering of useable

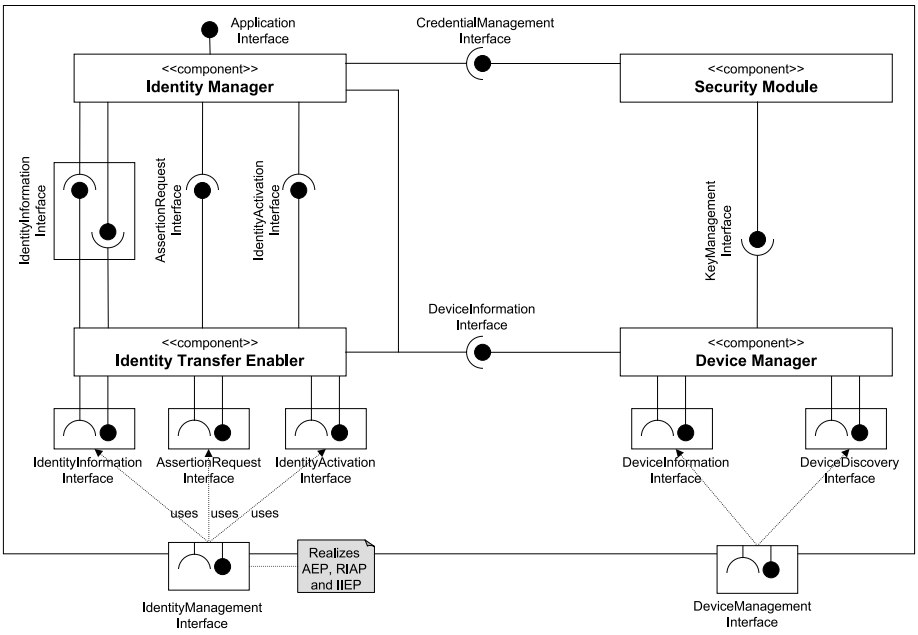


Figure 3: Component View

identities as well as the acquisition of information about available and active identities (R2, R3, R7).

Assertion Exchange: To establish a SP session on a device without an IdP session, we need to transfer assertions from one device to another. This is realized by the Assertion Exchange protocol (AEP) (see Section V-A) that allows requesting and obtaining tokens for a particular identity. The providing device checks the request against the configured policies that limit the usage of identities across devices. As an optional security mechanism, the user has to confirm the request on the providing device.

Remote Activation: If an IdP session for a dedicated identity cannot be established on one device, it has to be possible to activate this identity on another device. This is realized by the Remote Identity Activation Protocol (RIAP) (see Section V-A). Activation of identities on other devices requires that the requested device is in the proximity of the user and that the request is authorized.

Filtering of Identities: We assume that not each identity can be used on every device. First, specific credentials or specific authentication methods might be required to activate an identity that are not available on all devices. Second, the usage context of an identity might not be appropriate for a device (see Section II-C). Third, an identity should not be activatable on some devices due to security constraints, e.g. the security level is too low to use a dedicated identity.

Active Identities: The information on activated identities, i.e. an IdP session exists, should be available on other devices. With this information the number of active IdP sessions can be reduced and the burden for users to reauthenticate on another device is decreased. For the exchange of information related to identities, we have introduced the Identity Information Exchange Protocol (IIEP).

V. Architecture

The architecture realizes the three key concepts (see Section IV. Fig. 2 illustrates the interfaces between the devices that are part of a Virtual Device.

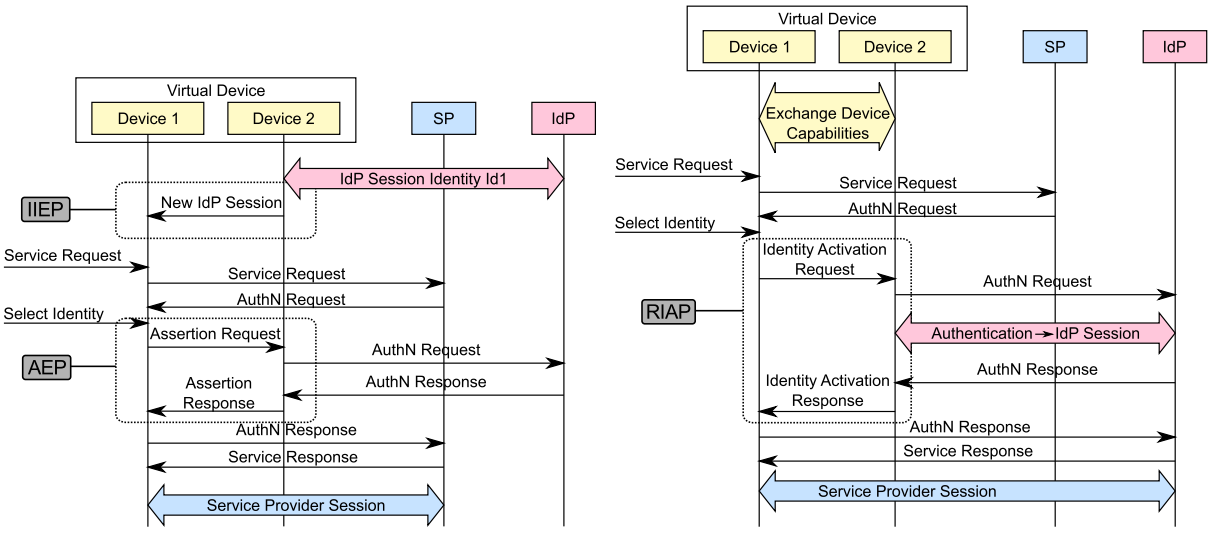
The Device Management interface realizes the Virtual Device concept, i.e. other devices are discovered, information about devices are exchanged and security associations are established. The Identity Management interface provides all functionality required to realize the Session Split and Multi-device IdM concept.

Fig. 3 provides a more detailed view on the components inside one device. It consists of the Identity Transfer Enabler, the Identity Manager, the Security Module and the Device Manager.

The functionality of the Device Manager, which realizes the Device Management interface has already been mentioned before. The Security Module provides a secure storage for certificates, private keys and credentials that have been obtained during the establishment of an IdP session. The realization of the security module is out of scope of this paper. In the following, we go into the details of the Identity Transfer Enabler and the Identity Manager, because both components realize in cooperation the Multi-device IdM concept.

A. Identity Transfer Enabler

The Identity Transfer Enabler implements the AEP, the RIAP and the IIEP.



(a) Assertion Exchange Protocol

(b) Remote Identity Activation Protocol

Figure 4: Message Flows within Virtual Device

AEP is used to obtain SP assertions from another device on which already an IdP session for the corresponding identity exists. Fig. 4a illustrates the corresponding message flow that is aligned with the Security Assertion Markup Language (SAML) protocols. It is assumed that the user has authenticated against the IdP on Device 2, thus an IdP session exists which allows the creation of tokens for SSO and that Device 1 is informed about the identity activation by means of IIEP. If the user switches to Device 1, requests a service and selects the same identity that is already active on Device 2, we are in the position to avoid an additional authentication procedure by exploiting the existing IdP session on Device 2.

The received Authentication Request (AuthNRequest) messages are encapsulated in an Assertion Request and forwarded from Device 1 to Device 2, which requests the needed assertion and returns it in the Assertion Response message to Device 1 that decapsulates the contained Authentication Response (AuthNResponse) and forwards it to Device 1. Based on the successful verification of the provided assertions, the SP grants access to the service and a SP session is established.

RIAP renders it possible to activate identities on remote devices (see Fig. 4b). This provides the opportunity to make use of secure devices for authentication or fulfill specific SP requirements regarding authentication. In addition, devices with acceptable user interfaces can be used for authentication to improve usability. Fig. 4b illustrates one scenario in which the Remote Identity Activation Protocol can be used. It is assumed that the devices have exchanged information about their capabilities. The SP requests a specific authentication method (e.g. authentication by means of a SIM card) in the AuthNRequest that cannot be met by Device 1 for the selected identity. Therefore, Device 1 triggers the activation of the identity by sending an Identity Activation Request to Device 2. The Identity Activation Request contains the AuthNRequest received from the SP. Subsequently, the user authenticates on Device 2 against the IdP and the required tokens are returned to Device 1 in order to establish the SP session.

B. Identity Manager

As the name indicates, the identity manager is responsible for managing the user's digital identities. This includes an interface for applications that can be used to request the activation of identities, functionality to select identities and a framework for the prefiltering of identities.

The filtering framework decides about the identities that can be used on a device. Fig. 5 illustrates the two step filtering process. In a first step, meta data on identities and meta data on devices is brought together with policies to create a list of identities that can be used on a device. The meta data on devices describes the authentication capabilities, the security level and the usage context. Meta data on identities contains supported authentication methods and security requirements for the device on which the identity should be used. This information might be provided by the IdP. In addition the user can add specific meta data, e.g. the usage context of the identity. Information about identities that are known on other devices is exchanged through the Identity Transfer Enabler.

Policies provide a flexible way to determine which identities can actually be used under which circumstances. Policies can be specified by the user itself or by system

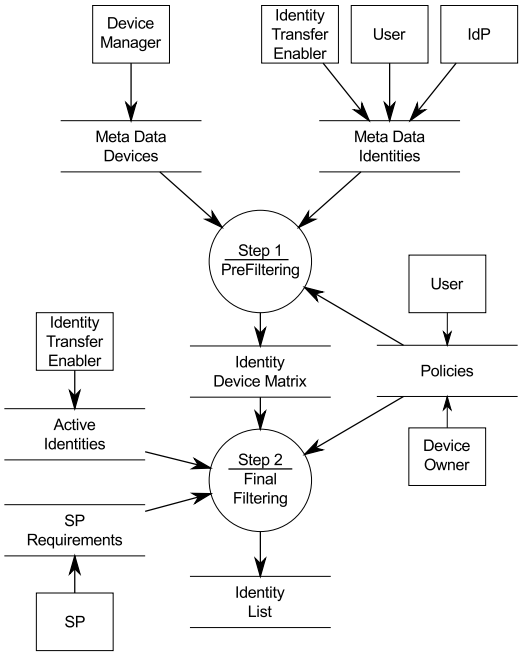


Figure 5: Identity Filtering Process

administrators of companies. We have selected eXtensible Access Control Markup Language (XACML) [15] as the policy language. In consequence we can differentiate three kinds of identities:

- Directly usable identities: This category comprises identities that can basically be used without any interaction with other devices. That means the device is secure enough, supports the authentication method needed to activate ($\rightarrow$ authentication against IdP) the identity and the usage context fits.
- Indirectly usable identities: Identities that can only be used with the support of another device that takes over the authentication against the IdP and provides the required assertions (c.f. RIAP).
- Unusable identities: All identities that cannot be used on a device are in this category. Reasons why an identity cannot be used on a particular device are among others an insufficient security level or an inappropriate usage context.

Since SP's requirements regarding authentication and necessary user data as well as the active identities are subject to dynamic changes, we introduced a second filtering step. The outcome is an identity list that is presented to users for identity selection. We currently do not consider privacy aspects in our filtering framework. In a third filtering step, we plan to use the metrics defined in [16] to consider privacy aspects.

VI. Implementation

We have implemented the architecture as a proof-of-concept for the Shibboleth IdM system. We have selected the Shibboleth IdM system due to the availability of the source code and prior experience with Shibboleth. It is basically possible to implement the same concept on top of OpenId.

Our testbed consisted of four machines, each running Ubuntu Linux 10.04 as depicted in Fig. 6. The IdP as well as the SP are running the Shibboleth IdM software [17] on top of the Apache webserver and TomCat application server, respectively. The available source code of the IdP and SP has not been modified. Since the focus of the proof-of-concept implementation is on the exchange of assertions and the remote activation of identities, we have restricted the Virtual Device concept to the necessary parts. That means we have no dynamic device discovery, i.e. every device knows all other devices in advance. Adding dynamic device discovery can be achieved by using the AllJoyn framework [18].

The establishment of security associations between devices has been realized by means of X.509 certificates that have been signed by a private key, which is dedicated to one Virtual Device. In addition, we maintain certificate revocation lists within a Virtual Device to cope with lost devices. Confidentiality-protected and authenticated channels between the devices are established using the TLS protocol [19].

So far the implementation effort has been limited to a few thousand lines of code, which confirm the principle feasibility of the designed concepts. Currently we are working on the integration of our concepts into web browsers based on the SAML Enhanced Client or Proxy Profile (ECP) [20].

VII. Security Evaluation

With the Virtual Device concept we have introduced additional interfaces between devices that are subject to attacks. Therefore, we considered security during the complete design process. During the requirement specification phase we specified security requirements based on an asset and threat analysis. We used the results to extend the functional requirements in order to introduce appropriate countermeasures.

Finally, we evaluated the designed architecture against the requirements. In addition we created extended use cases (so called misuse cases), how the system might be misused. In the following we focus on our attacker model, give an overview on the considered attacks and exemplify the methodology along one sample attack. Finally we give a short overview on introduced countermeasures.

A. Attacker Model

We can differentiate two categories of attackers: Internal and External Attackers

Internal attackers are the user, the IdP and the SP. It is essential for IdM that the user trusts at least the IdP.

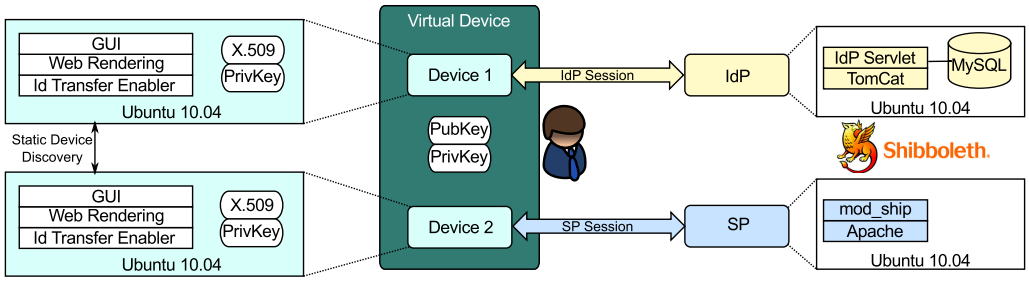


Figure 6: Conceptual view on prototype

Therefore, the IdP or an SP are not considered as attackers. Moreover, we want to provide the user with a system that assists him with authentication across all his devices and we do not want to protect the user from himself. Thus, the consideration of internal attackers was out of scope. Our focus is on *external attackers*, i.e. all entities that are not directly participating. We focused in particular on attacks that result from the introduction of the Virtual Device concept and the communication between the devices.

B. Attack Overview

We identified three different protection goals that an external attacker might consider: Authenticity, Privacy and Availability. Figure I provides an overview of the attacks with the corresponding countermeasures.

Attack	Attacker	Countermeasures
Authenticity: Illegal Service Consumption	External	Authentication of devices Encrypted information exchange between devices Manual confirmation of requests
Privacy: Observation of Virtual Device	External	Privacy-protecting service discovery Encrypted information exchange between devices
Privacy: Observation of SP and IdP Sessions	External	Encrypted information exchange between devices Encryption of stored information within Virtual Device
Availability: Interruption of Virtual Device Operation	External	Fallback to independence of Virtual Device

Table I: Attack Overview

C. Attack Trees

For the modeling of attacks we used the attack tree methodology [21]. Fig. 7 illustrates how an attacker might consume a service on behalf of a user. Each path from the root of the graph to the leaves represents a potential attack. The highlighted attack paths represent potential attacks introduced with the Virtual Device. For example, the attacker could establish a new service session by requesting a SP Token from one of the devices that is part of the Virtual Device. We considered all paths and confirmed that appropriate security mechanisms are in place.

D. Countermeasures

In order to avoid unauthorized identity activation and requests for SP tokens from other devices we have introduced the following mechanisms. First, all devices within a Virtual Device have to authenticate against each other. This avoids that foreign devices are considered as part of the Virtual Device. Second, the communication channel between devices of a Virtual Device is encrypted. Third, activating identities and requesting SP tokens from another device has to be confirmed by the user in order deal with lost and stolen devices. Finally, we assume that all user devices provide all locking functionality to prevent unauthorized usage.

VIII. Related Work

A. Virtual Devices and Personal Networks

The concept of Virtual Devices or Personal Networks is well known in literature [13], [14], [22]. Several devices belonging either to one user or to trusted parties are cooperating to achieve a common goal. Among the goals are:

- Network access: One device acts as a gateway and provides global connectivity to other devices. Hereby, the selection of the gateway and the corresponding interface is challenging [12], [23].
- Data provisioning: Users want access to their personal data from all devices. In particular for copyright protected content, content owners want to limit access by digital rights management systems. Several solutions have been standardized [24] or proposed to enable access on all user devices [25], [26].
- Capability sharing: The devices being part of a Virtual Device are heterogeneous regarding their capabilities (e.g. display size, computing power, ...). From the user perspective the most benefit can be obtained, if the devices share their capabilities [14], [27]. This includes the distribution of multimedia session across several user devices [28], [29] but also the relaying of computing intensive tasks to powerful machines [30].
- Context Management: All devices belonging to a Virtual Device can cooperate to capture and process context information. Reasoned by the heterogeneity of devices and thus the available sensors, much more context information can be gathered [31].

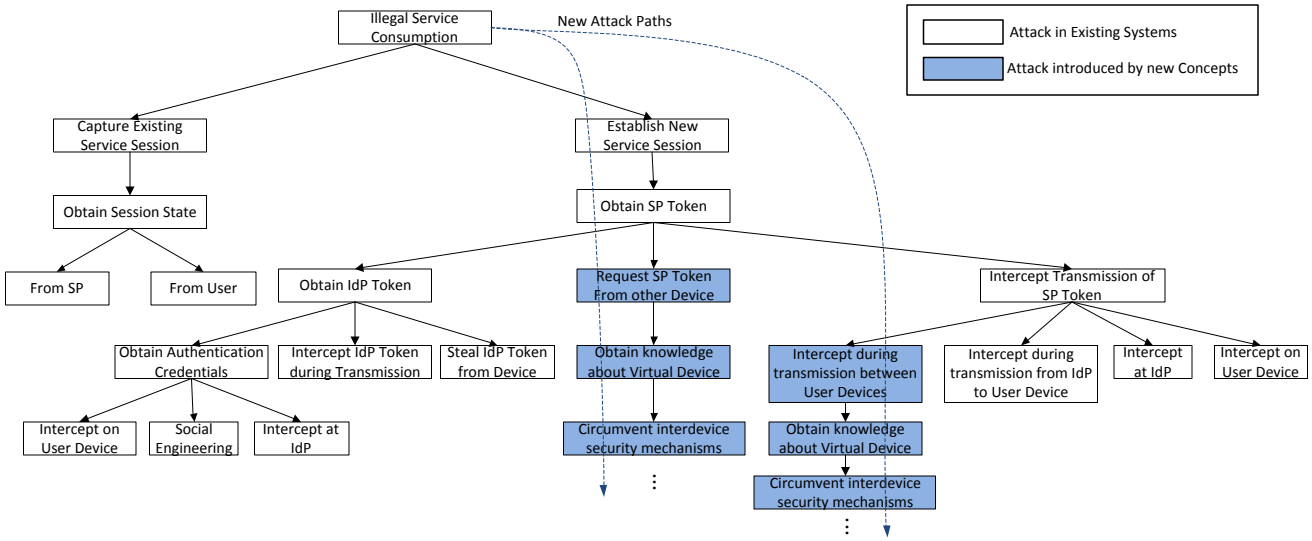


Figure 7: Attack tree for illegal service consumption

To achieve these goals all solutions have to address several challenges. These challenges are device discovery, trust establishment between devices and the exchange of device capabilities. Several solutions for device discovery have been proposed, e.g. the Service Discovery Protocol (SDP) provided by Bluetooth, the Service Location Protocol (SLP) [32] or DNS-based service discocery (DNS-SD) [33]. We assume that any protocol is adequate within our architecture.

Trust between devices can be established by device pairing. As a result of device pairing the devices have established security associations. Surveys of device pairing methods are provided in [34], [35]. Basically our simple approach for establishing trust between devices can be replaced by a more sophisticated method.

For the description of device capabilities, several standards exist. CC/PP [36] as well as OMA User Agent Profile [37] focus on the adaptation of content for optimal user experience on user devices. An alternative initiative is WURFL [38], which provides open access to device descriptions. All possibilities are not adequate for our scenarios, since they neither describe the usage context of a device nor its security capabilities.

B. Identity Management

Identity Management is a very comprehensive topic and has different facets. Single Sign-On, Attribute Provisioning and Single Log-Off increase usability and security from the user's perspective. Such capabilities are provided by identity management systems like OpenId [4], Shibboleth [5], Liberty Alliance [6] or Microsoft Cardspace [7].

Another facet of identity management is privacy protection. A SP should only obtain as much information about the user as actually necessary [39]. Based on the requirements of the SP, the user is able to select one of his "partial identities" [39], [40] to restrict the available information. The concept of "partial identities" is also known as "virtual identities" [41]. Another approach to protect the privacy of users is based on anonymous credentials [42]. Since, our focus is not on privacy protection, we do not go into more detail.

C. Multi-Device Authentication and Identity Management

Several solutions exist that provide users the possibility to consume services from different devices. We can classify existing solutions into three categories.

Personal Authentication Devices: Several solutions introduce dedicated authentication devices [43], [44]. Wong et.al. [43] have been the first, to the best of our knowledge, that introduce the concept of Personal Authentication Devices, which are used to authenticate against services independent of the actually used device. Even if they take multiple user-devices into account, no solution regarding SSO and federation is provided. Moreover, they do not provide sufficient usability, because the user has to manually enter a PIN. Consequences of the resulting trust model are discussed in [45]. Corisecio [46] provides an extension of the Personal Authentication Device based on Microsoft Cardspace. The developed solution stores all identity cards of a user on a mobile device and makes these cards available to other devices.

Distribution of Credentials: An alternative solution is the distribution of credentials to all devices owned by the user. A common solution is the distribution and synchronization of password stores across user devices. Solutions like Xmarks [47] store user passwords on a central server and allow the retrieval to all user devices. Such a solution has severe security drawbacks. Even if all passwords are stored encrypted on a central server, they are exposed to security threats like brute-force attacks.

Requesting Credentials: If a device can request the required credentials on demand from another device it is

grouped into this category. With session mobility in mind, Liberty Alliance has proposed a solution to transfer credentials on demand between devices [48]. Combined with the transfer of the application context, which is required to enable session mobility, so called endpoint references can be transferred. Endpoint references allow the creation of an additional SAML assertion for service authentication. This solution assumes a preestablished trust relationship between the participating devices, but specifies no additional details. To the best of our knowledge, further consideration of these initial concepts has stopped. Recently, an extension to OpenAuth has been proposed [11]. The proposal addresses authentication on devices with limited input devices (e.g. no keyboard to enter password). Hereby, the authentication is performed on a more powerful device and authorization tokens are transferred. Trust between the devices is manually established on demand, i.e. an identifier is displayed on the limited device and has to be manually entered on the more powerful device. In addition our solution falls into this category. In contrast to [11] we consider preestablished trust relationships between devices. Moreover we consider different identities as well as security levels of devices to authorize the usage of identities across user devices.

D. Security Level of Devices

There is no absolute measure for the security level of a device. By means of trusted computing we can confirm that a system has neither been modified [49] nor its configuration files have changed [50].

An alternative is the determination of the installed software and the corresponding version. Originally introduced for network access control, we could basically reuse the daemons for network endpoint assessment [51].

IX. Conclusion

We presented an architecture that integrates all devices of a user and provides ubiquitous user authentication across devices by using identity management concepts. With such an approach we are able to demonstrate usability and security improvements by taking the diverse characteristics of devices and identities into account. Potential use of the architecture has been motivated by usage scenarios.

The implemented prototype confirmed the feasibility of the developed concepts on the basis of the Shibboleth IdM system. We considered security during all stages of the design and confirmed by the conducted security evaluation that we can cope with the challenges introduced by the additional complexity of several devices.

Currently, we are evaluating the performance of our architecture. In particular we are interested in the impact of the Virtual Device concept on the number of required user authentication procedures. Since, we do not consider real life studies as feasible, we extend the methodology introduced in [52].

References

[1] M. Weiser, "The Computer for the 21st Century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, pp. 3–11, July 1999.

[2] T. Hunt, "A brief Sony Password Analysis," http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html, June 2011.

[3] C. Gillespie, "Character Occurence in Passwords," http://csgillespie.wordpress.com/2011/06/16/character-occurrence-in-passwords/, June 2011.

[4] D. Recordon *et al.*, "OpenID Authentication 2.0 - Final," December 2007.

[5] T. Scavo *et al.*, "Shibboleth Architecture Technical Overview, Working Draft 02,," June 2005.

[6] J. Tourzan *et al.*, "Liberty ID-WSF Web Services Framework Overview, Version 1.1."

[7] V. Bertocci *et al.*, *Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities*. Addison-Wesley Longman, 2008.

[8] M. Lischka *et al.*, "SWIFT Deliverable D502 - SWIFT Scenarios, Use Cases and Business Models," 2008.

[9] J. Jähnert *et al.*, "Description of DAIDALOS II Scenario Design Report," www.ict-daidalos.org, September 2008.

[10] U. Jendricke and D. Gerd tom Markotten, "Usability meets Security - the Identity-Manager as your Personal Security Assistant for the Internet," in *Computer Security Applications, 2000. ACSAC '00. 16th Annual Conference*, dec 2000, pp. 344 –353.

[11] D. Recordon and B. Goldman, "OAuth 2.0 Device Profile – IETF Internet Draft draft-recordon-oauth-v2-device00.txt," 07 2010.

[12] R. Atkinson, J. Irvine, J. Dunlop, and S. Vadgama, "The Personal Distributed Environment," *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 62 –69, april 2007.

[13] D. Calin, A. R. McGee, U. Chandrashekhar, and R. Prasad, "MAGNET: An Approach for Secure Personal Networking in Beyond 3G Wireless Networks," *Bell Labs Technical Journal*, vol. 11, no. 1, pp. 79–98, 2006.

[14] R. Y. Fu *et al.*, "A Framework for Device Capability on Demand and Virtual Device User Experience," *IBM Journal of Research and Development*, vol. 48, no. 5, pp. 635–648, 2004.

[15] T. Moses *et al.*, *eXtensible Access Control Markup Language (XACML) V2.0*, OASIS Std.

[16] M. Neubauer, "Modelling of Pseudonymity under Probabilistic Linkability Attacks," in *International Symposium on Secure Computing (SecureCom09)*, August 2009.

[17] "Shibboleth IdM Software Version 2.2," http://shibboleth.internet2.edu/, 2011.

[18] "AllJoyn Android NDK - Developer Guide," www.alljoyn.org, February 2011.

[19] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," RFC 4346, IETF, Apr. 2006.

[20] J. Hughes *et al.*, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS standard, March 2005.

[21] B. Schneier, "Attack Trees," *Dr Dobb's Journal*, vol. 24, no. 12, p. 1, December 1999.

[22] I. Niemegeers and S. Heemstra de Groot, "From Personal Area Networks to Personal Networks: A User Oriented Approach," *Wireless Personal Communications*, vol. 22, pp. 175–186, 2002.

[23] U. Javaid, D.-E. Meddour, T. Rasheed, and T. Ahmed, "Mobility Management Architecture for Personal Ubiquitous Environments," in *PIMRC 2008*, sept. 2008, pp. 1 –5.

[24] "Open Mobile Alliance – DRM Architecture Version 2.2," March 2011.

[25] S. Sovio, N. Asokan, and K. Nyberg, "Defining Authorization Domains Using Virtual Devices," in *SAINT'03 Workshop*. Washington, DC, USA: IEEE Computer Society, 2003, p. 331.

[26] P. Koster, J. Montaner, N. Koraichi, and S. Iacob, "Introduction of the Domain Issuer in OMA DRM," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, jan. 2007, pp. 940 –944.

[27] M. Schuster, A. Domene, R. Vaidya, S. Arbanowski, S. M. Kim, J. W. Lee, and H. Lim, "Virtual Device Composition," in *Proc. Eighth Int. Symp. Autonomous Decentralized Systems ISADS '07*, 2007, pp. 270–278.

[28] J. A. Tuijn and D. Bijwaard, "Spanning a Multimedia Session across Multiple Devices," *Bell Labs Technical Journal*, vol. 12, no. 4, pp. 179–193, 2008.

[29] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Ubiquitous Device Personalization and Use: The next Generation of IP Multimedia Communications," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 2, p. 12, 2007.

[30] K. Yang, S. Ou, and H.-H. Chen, "On Effective Offloading Services for Resource-constrained Mobile Devices Running Heavier Mobile Internet Applications," *Communications Magazine, IEEE*, vol. 46, no. 1, pp. 56 –63, january 2008.

[31] M. Bauer *et al.*, "Context Management Framework for MAGNET Beyond," in *Open international workshop on capturing context and context aware systems and platform*, 2006.

[32] E. Guttman *et al.*, "Service Location Protocol, Version 2," RFC 2608, Internet Engineering Task Force, Jun. 1999.

[33] S. Cheshire and M. Krochmal, "Dns-based Service Discovery – IETF Draft draft-cheshire-dnsext-dns-sd-10.txt," February 2011.

[34] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, "A Comparative Study of Secure Device Pairing Methods," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 734 – 749, 2009.

[35] J. Suomalainen *et al.*, "Standards for Security Associations in Personal Networks – A Comparative Analysis," *Int. J. Secur. Netw.*, vol. 4, no. 1/2, pp. 87–100, 2009.

[36] "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0 – W3C Rec." January 2004.

[37] "Open Mobile Alliance – User Agent Profile 2.0," Feb. 2006.

[38] "Wireless Universal Resource File (WURFL) – http://wurfl.sourceforge.net/," 2011.

[39] S. Clauss and M. Köhntopp, "Identity Management and its Support of Multilateral Security," *Comput. Netw.*, vol. 37, pp. 205–219, August 2001.

[40] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Mobile Identity Management," in *Inproceedings of Ubicomp workshop*, 2002.

[41] A. Sarma *et al.*, "Virtual Identity Framework for Telecom Infrastructures," in *Wireless Personal Communications*. Netherlands: Springer, February 2008.

[42] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-transferable Anonymous Credentials with Optional Aonymity Revocation," vol. 2045, pp. 93–118, 2001.

[43] R. Wong *et al.*, "Polonius: An Identity Authentication System," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1985, pp. 101–107.

[44] A. Pfitzmann, B. Pfitzmann, M. Schunter, and M. Waidner, "Trusting Mobile User Devices and Security Modules," *Computer*, vol. 30, no. 2, pp. 61 –68, feb 1997.

[45] A. Josang *et al.*, "Trust Requirements in Identity Management," in *Proc. of Australasian Workshop on Grid Computing and e-research*, Darlinghurst, 2005, pp. 99–108.

[46] Corisecio, "Mobile Cardspace." [Online]. Available: http://www.corisecio.com

[47] XMarks, "Secure Password Sync," http://www.xmarks.com, 2009.

[48] P. Madsen, "Liberty ID-WSF Multi-Device SSO Deployment Guide," 10 2008.

[49] E. Gallery and C. J. Mitchell, "Trusted Computing: Security and Applications," *Cryptologia*, vol. 33, pp. 217–245, 2009.

[50] R. Sailer *et al.*, "Attestation-based Policy Enforcement for Remote Access," in *Proceedings of CCS 2004*, New York, 2004, pp. 308–317.

[51] P. Sangster *et al.*, "Network Endpoint Assessment (NEA): Overview and Requirements," RFC 5209 (Informational), June 2008.

[52] M. Barisch, "Modelling the Impact of Virtual Identities on Communication Infrastructures," in *Proceedings of the 5th ACM workshop on Digital Identity Management*. New York, NY, USA: ACM, 2009, pp. 45–52.