Special Issue

Detlef Hühnlein*, Tobias Wich, Johannes Schmölz, and Hans-Martin Haase

The evolution of identity management using the example of web-based applications

Abstract: The typical identity management (IdM) techniques used in web-based applications are about to change from application-specific means for identification, authentication and authorization towards the support of standardized, secure and privacy friendly mechanisms for Single Sign-On (SSO). In this paper we outline the different phases of this evolution, which started with the introduction of standardized interfaces for authentication and authorization and allowed to shift these sensitive tasks from the application towards the web application server. In a second phase the interfaces were extended to support authentication and authorization in distributed systems and feature SSO-techniques. The third phase adds identification and aims at providing more security for distributed authentication infrastructures and finally there is a trend towards providing more privacy friendly mechanisms for identity management in the future.

Keywords: ACM CSS → Security and privacy → Web application security, ACM CSS → Security and privacy → Security services, SSO, SAML, OpenID, OAuth 2, Attribute-based Credentials, eID.

DOI 10.1515/itit-2013-1036
Received October 7, 2013; accepted April 4, 2014

1 Introduction

The popularity of web-based applications is continuously increasing, as they do not require to download, install and update application-specific client software, but allow to access and use a centrally managed web-application using a simple web browser instead. Furthermore it is possible to access the same application and data from a variety of desktop or mobile client systems from anywhere and at any time. This enables seamless access to applications within and across organizational boundaries, which boosts productivity and allows to reduce costs. In particular there are economic benefits, if it is possible to use standardized services offered by specialized cloud service providers [6, 38], as this removes the need for upfront investments.

On the other hand this flexibility and openness imposes various challenges for the involved identity management (IdM) techniques used for identification, authentication and authorization of users and services. Against this background these techniques have been subject of an evolution, which started with the introduction of standardized interfaces for authentication and authorization, which allowed to shift these security sensitive tasks from the web-based application to the web application server. This *first* phase is addressed in Section 2.1.

In the *second* phase the previously local interfaces were extended to support distributed authentication systems and Single Sign-On (SSO) as explained in Section 2.2 and depicted in Figure 1. In this setting a web-application (Service Provider, *SP*) delegates the authentication procedure to a specialized authentication service (Identity Provider, *IdP*), which performs the authentication on behalf of the web-application and exchanges the user's credential C_U for a session credential C_S, which finally is presented to the *SP* using some federation protocol, such as the Security Assertion Markup Language (SAML), OpenID, WS-Trust/WS-Federation or OAuth for example.

In this situation an adversary may either try to circumvent the authentication process or steal and misuse the session credential C_S, which grants access to the protected resource. The *third* evolution phase as discussed in Section 2.3 aims at countering these attacks by providing means for stronger authentication, identification using national ID cards for example and strengthening the federation protocols using cryptographic bindings, which prevent that the session credentials can be stolen and misused by an attacker.

Apart from all the benefits centralized IdM systems may provide, there is an amplification of the risk associated with a potential security breach and the centralization imposes additional threats to the privacy of users,

***Corresponding Author: Detlef Hühnlein,** ecsec GmbH, Michelau, e-mail: detlef.huehnlein@ecsec.de
Tobias Wich, Johannes Schmölz, Hans-Martin Haase: ecsec GmbH, Michelau

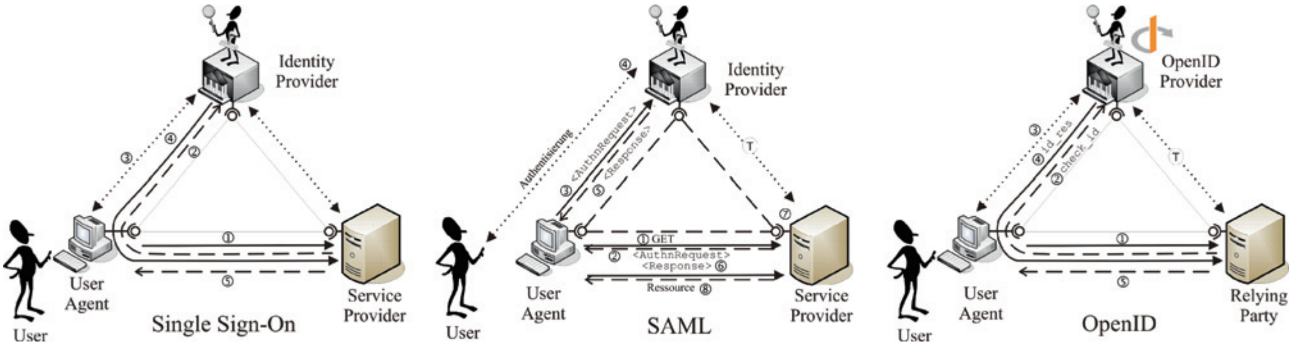

Figure 1: Single Sign On, SAML and OpenID overview.

which may become a major driver for the *fourth* evolution phase in identity management as explained in Section 2.4. This era, which has not yet fully entered practice, involves the introduction and widespread deployment of privacy enhancing technologies, such as Attribute Based Credentials (ABC) based on Idemix [7] or U-Prove [26].

2 IdM-Evolution

2.1 Local authentication services

Despite the various deficiencies (cf. [11, 24]) it is still common to use password-based authentication mechanisms for the login to web-based services and as humans are often not capable to memorize a large number of different passwords with high entropy it is unfortunately also common practice to re-use passwords, which in turn leads to additional risks [21]. A promising and increasingly popular approach to solve this problem is to employ multifactor and/or out of band authentication. However integrating smart cards, smart phones, One Time Password (OTP) tokens, certificates and the like is more complicated and more expensive than integrating a simple username and password based authentication. In order to minimize the efforts related to supporting various authentication technologies it is advisable to decouple the authentication modules from the application components using appropriate interfaces such as the Java Authentication and Authorization Service (JAAS) [35] or its successor the Java Authentication Service Provider Interface for Containers (JASPI) [22] for example. This allows to shift the security sensitive authentication and authorization processes from the web-based applications to the web application server, which is usually easier to safeguard and maintain than multiple application-specific authentication modules. The management of user accounts in such a setting is usually performed by means, which are specific to the web-application or authentication module under consideration.

2.2 Single sign-on

In order to improve the user experience and to support authentication and authorization in distributed systems, the previously local interfaces have been extended to support distributed processing and Single Sign-On (SSO).

As depicted in Figure 1, three parties are involved in such a federated SSO-scenario: The User (U) is equipped with a User Agent (UA) and wants to access a protected resource offered by the Service Provider (SP) in step (1). Instead of directly authenticating U, the SP delegates the authentication procedure to the specialized Identity Provider (IdP) in step (2), which performs the authentication in step (3) on behalf of SP and exchanges the User's credential C_U for a session credential C_S, which is presented to the SP in step (4) using an appropriate federation protocol, such as Kerberos, SAML, OpenID, WS-Trust/WS-Federation or OAuth for example. The SP in turn is validating the session credential C_S and finally delivers the protected resource in step (5) to the UA in case of success.

As discussed in the following there are different protocol variants, which map to the abstract SSO scheme depicted in Figure 1. The management of user accounts in this phase is either implicitly performed using the federation protocol under consideration or explicitly by additional provisioning protocols such as the Service Provisioning Markup Language (SPML) [10] or the System for Cross-domain Identity Management (SCIM)[1].

[1] See http://simplecloud.info.

2.2.1 Kerberos

The Kerberos protocol was designed to work in open (unprotected) networks for the purpose of verifying the identity of Principals. Principals are entities in the network which can be a User (U) with a User Agent (UA) or a server, which may act as Service Provider (SP). This protocol exchanges ASN.1 based data structures and was probably the first standardized SSO protocol. It is widely used in intranet and enterprise scenarios as it is the standard authentication mechanism in Microsoft products since Windows 2000 and plays an important role for authentication against infrastructure components, such as domain controllers. The basic infrastructure of a Kerberos network consists of Principals like UA and SP and an IdP, which comprises an Authentication Server (AS) and a Ticket-Granting Server (TGS). The process flow is roughly as follows: In order to access a protected resource, the UA requests a ticket-granting-ticket from the AS which is only issued after a successful authentication. This ticket-granting-ticket is now used to get tickets for the several services offered in the network. If a User U with a User Agent UA wants to access some resource at the SP he requests an access ticket for this server by sending the ticket-granting-ticket to the ticket-granting server which issues the access ticket for the SP. While Kerberos is not native to the web, the widely supported negotiation mechanism defined in RFC 4559 allows to use Kerberos authentication for web-based applications.

2.2.2 SAML

The Security Assertion Markup Language (SAML) is a family of standards, which has been developed by the OASIS Security Services Technical Committee and defines the syntax and semantics for XML-encoded assertions about authentication, attributes and authorization together with related protocols that convey such assertions and the binding of these protocols to various transfer protocols. The different versions of SAML have been influenced by previous work at IETF (cf. RFC 2903) and projects like the Liberty Alliance and Shibboleth. The current version of SAML is version 2.0 and comprises various parts, which define Assertions and Protocols, Bindings, Profiles, Metadata, Authentication Contexts, Conformance Requirements, a Glossary and last but not least Security and Privacy Considerations.

As depicted in Figure 1, the User U sends a resource request to the SP, which in step (2) returns an XML-based `AuthnRequest`, which initiates the authentication at the IdP in step (3). The IdP authenticates the User and returns a corresponding `Response` to the requesting SP in step (4). In case of success the `Response` contains a signed `Assertion`, which in turn may contain identity attributes of the user and details with respect to the performed authentication.

There are early proposals for the combination of SAML and Kerberos [13] and based on the recent SAML V2.0 Kerberos Web Browser SSO Profile it is possible to use Kerberos authentication in SAML-based SSO infrastructures.

The security of SAML v1.0 was analyzed in [16] and the discovered flaws led to recommendations in version 2.0 of SAML (cf. [23]). Additional vulnerabilities of the artifact profile have been addressed in [17]. A security analysis for a Liberty-enabled client can be found in [28].

A general treatment of security aspects related to the current version of SAML may be found in [19]. A formal security analysis of the Web Browser SSO Profile in SAML 2.0 only appeared fairly recently in [1] and revealed a flaw in the SAML-implementation of Google Apps. Other steps towards providing security proofs for browser based protocols can be found in [14, 18, 39]. In [31] it was shown that many SAML implementations were susceptible against signature-wrapping attacks.

2.2.3 OpenID

OpenID is a lightweight SSO protocol, whose process flow is depicted in Figure 1. In step (1) the User requests a resource via his UA at the SP, which is called Relying Party in OpenID terminology. If the User already provided the address of his OpenID provider as identifier, the SP just needs to redirect the UA in step (2) to the OpenID provider (IdP) with an HTML form encoded `checkid` message in order to request the authentication there. Otherwise the SP has to discover the IdP by using Yadis, XRI and XRDS or a similar discovery method [29]. After the authentication in step (3) the IdP issues an `id_res` authentication token and redirects the UA back to the SP in step (4), where the authentication token is verified and finally access is granted in step (5) in case of success. The security of OpenID has been analyzed in [25, 32, 37] for example and it has been shown that a naive implementation of the OpenID-protocol may be vulnerable to Man-in-the-Middle attacks as well as Parameter Injection and Parameter Forgery attacks to name a few.

2.2.4 OAuth 2

OAuth 2 (RFC 6749) is the SSO specification which especially aims at the modern web and mobile market. It is the successor of the first version of OAuth (RFC 5849) and tries to make the integration for the SP easier and thereby less error prone as stated on the official website[2]. Without any specific extensions it only uses the security provided by the Transport Layer Security (TLS) protocol instead of special client side cryptography.

While OAuth 2 is primarily an authorization protocol, it can also be used for authentication and SSO purposes, as the big web companies like Google and Facebook demonstrate. A closer look at the use of OAuth 2 at these companies reveals that their implementations closely follow the OpenID Connect[3] specification. One can hope that when OpenID Connect becomes more mature, the different OAuth 2 SSO solutions will be replaced by OpenID Connect solutions and provider interoperability, as anticipated by SAML, will also come to the OAuth 2 world.

OAuth 2 provides two main modes of operation or flows: the "server-flow" (or Authorization Code Grant) and the "client-flow" (or Implicit Grant). While the first one is used for web-applications which receive the access token by their server side logic, the client-flow is for JavaScript-based applications, which run completely in the browser. The security of OAuth 2 has been analyzed with formal methods in [9], practical security aspects of OAuth 2 have been studied in [33, 34] and a more systematic treatment can be found in RFC 6819. Similar as with OpenID above, the security of an OAuth 2-based system highly depends on details of the implementation and naive implementations may be attacked by Cross-Site-Scripting (XSS), Cross-Site-Request-Forgery (CSRF) and Man-in-the-Middle (MitM) attacks for example. Furthermore it seems to be worthwhile to point out that OAuth 2 as of today only uses "bearer tokens" and more secure bindings are still in a rather early stage of development [36].

Token based SSO systems and OAuth 2 in particular facilitate different authentication approaches in web-applications as it has been pointed out by Alberto Pose[4]. By using access tokens instead of session cookies for the authorization of service requests, some of the problems that are associated with OAuth 2 are solved, or at least become simpler to handle. Besides providing protection against CSRF attacks, cross domain requests are possible without workarounds and most importantly a stateless style, the basic principle behind REST services, is encouraged. However the flexibility regarding cross domain requests might expose new kind of vulnerabilities in future web-applications.

2.3 Secure SSO

In order to enhance the security of SSO-systems there are two independent aspects to consider. On the one hand side one needs to enhance the security of the authentication step and provide "strong authentication" as discussed in Section 2.3.1. On the other side one needs to improve the security of the federation protocol and employ strong bindings as discussed in Section 2.3.2.

If one uses identity cards issued by governments for strong authentication, it is also possible to support the identification of users and the provision of identity attributes with high assurance levels. Based on these identity attributes it may in turn be possible to support the implicit provisioning of user accounts and corresponding access rights based on these attributes and hence one may omit explicit means for the provisioning of user accounts.

2.3.1 Strong authentication

For the authentication of persons, devices and services there are countless cryptographic protocols [3]. Even in a very abstract perspective one may distinguish between different standardized mechanisms (see ISO/IEC 9798 and ISO/IEC 11770 for example) and the scenario becomes more complex if one considers the concrete realization of the personal security environment of a user, which may support various factors, such as knowledge, possession, biometrics, location [2] or relationships [4] for example. Similar to the separation of authentication modules in web application servers (cf. Section 2.1) one may use interfaces, such as JCE/JCA, ISO/IEC 24727, PKCS#11 or Microsoft's CSP for example, to decouple cryptographic mechanisms and tokens from application-specific modules. By using XML-based CardInfo-files proposed in [20] and standardized in ISO/IEC 24727-3 together with appropriate authentication services as developed in the SkIDentity[5] project it is fairly easy to support a wide range of smart cards for strong authentication. In this case one may also use national identity tokens in order to support the identification

[2] See http://oauth.net/2/.

[3] See http://openid.net/connect/.

[4] See http://blog.auth0.com/2014/01/07/angularjs-authentication-with-cookies-vs-token/.

[5] See http://skidentity.de.

of users and provide identity attributes with high assurance levels according to ISO/IEC 29115.

2.3.2 Secure binding

Analyzing the list of potential attacks against SAML, OpenID and OAuth 2 reveals that many threats are due to the missing cryptographic binding between the federation protocol messages and the underlying transport protocol. The Man-in-the-Middle (MitM) attack visualized in [12, Figure 5] exploits this weakness.

As of today there have been different proposals for binding SAML and similar federation protocols to the underlying TLS channel in order to safeguard against MitM-attacks:

TLS-Federation. In this approach [5], the SAML assertion is sent inside a short-lived X.509 client certificate. The SAML assertion thus may replace other identification data like distinguished names and the certificate has the same validity period as the SAML assertion.

Strong Locked SOP. Here [30], the client is strengthened to make reliable security decisions. This is done by using the servers public key as a basis for decisions of the Same Origin Policy (SOP), rather than the insecure Domain Name System. The certificate verification procedure specified in BSI-TR-03112-7 (v1.1.2) in which it is checked that the hash of the X.509 certificate obtained during the TLS handshake is included in the Card-Verifiable-Certificate (CVC) may be seen as a variant of this approach.

Holder-of-Key Binding. This approach also uses TLS with client authentication, but the client certificate does not transport any authorization information. Instead, the SAML token is bound to the public key contained in this certificate, by including this key in a Holder-of-Key (HoK) assertion. The security of this approach has independently been analyzed in [15].

TLS-Channel Bindings. Finally, the generic channel binding mechanism sketched in RFC 5056 can be applied to TLS, SAML and various other Single Sign-On solutions.

2.4 More privacy

It is a known fact that the web's large SSO providers, like Google and Facebook for example, analyze user behavior in at least a pseudonymous way and it would be just a little additional step to map these pseudonyms to individual users. Furthermore against the background of the recent news with respect to PRISM and TEMPORA it is hardly possible to exclude the possibility that secret government services may already create such user profiles. Therefore it is desirable to provide strong mechanisms for identification and authentication, which do not require to disclose more identity attributes than absolutely necessary and allow pseudonymous and when possible anonymous access to web-based services.

Idemix[6] [7, 8] and U-Prove[7] [26, 27] are two promising approaches to realize privacy friendly Attribute Based Credentials (ABC). Both approaches use sophisticated cryptographic techniques such as zero-knowledge proofs for the selective disclosure of identity attributes. This means that a user is equipped with a cryptographic credential, which allows to prove certain predicates about her identity. Instead of revealing her date of birth, she is able to prove that her age is in a certain interval for example. While both systems provide similar general features, the Idemix system also supports a very strong notion of unlinkability and hence allows to design very privacy friendly identity management systems. While both approaches are based on not yet standardized and not yet widely used cryptographic techniques, the EU-funded ABC4Trust[8] project has unified the high level access to these advanced technologies and there is ongoing research in the FutureID[9] project, which aims at integrating Idemix with main stream eID technology.

3 Summary and outlook

The present paper has shown that the Identity Management techniques have gone through different phases of evolution, which comprise the separation of authentication modules (1), Single Sign-On (2), the enhancement of security by using strong authentication techniques and cryptographic bindings for authentication tokens (3) and finally the enhancement of privacy by using sophisticated Attribute Based Credential technologies (4), which feature anonymous access to web-based services. As the different evolution phases have been addressed by various research projects, such as ABC4Trust, FutureID and last but not least SkIDentity one may expect that the corresponding innovations may soon enter practice and make real world

[6] See http://www.zurich.ibm.com/security/idemix/.

[7] See http://research.microsoft.com/en-us/projects/u-prove/.

[8] See https://abc4trust.eu.

[9] See http://futureid.eu.

web-applications a little more usable, secure and privacy friendly.

References

1. A. Armando, R. Carbone, L. Compagna, J. Cuellar, and L. Tobarra. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. Formal Methods in Security Engineering 2008, ACM-Press, 2008.
2. E. Bertino and M. S. Kirkpatrick. Location-aware authentication and access control. In Irfan Awan, Muhammad Younas, Takahiro Hara, and Arjan Durresi, editors, *AINA*, pages 10–15. IEEE Computer Society, 2009.
3. C. Boyd and A. Mathuria. *Protocols for authentication and key establishment*. Springer-Verlag, 2003.
4. J. G. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung. Fourth-factor authentication: somebody you know. In *ACM Conference on Computer and Communications Security*, pages 168–178, 2006.
5. B. P. Bruegger, D. Hühnlein, and J. Schwenk. TLS-federation – a secure and relying-party-friendly approach for federated identity management. In *Proceedings of BIOSIG 2008: Biometrics and Electronic Signatures*, volume 137 of *Lecture Notes in Informatics (LNI)*, pages 93–104. GI-Edition, 2008.
6. R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616, 2009.
7. J. Camenisch and T. Groß. Efficient attributes for anonymous credentials. In *ACM Conference on Computer and Communications Security 2008*, pages 345–356. ACM-Press, 2008.
8. J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Vijayalakshmi Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 21–30. ACM, 2002. http://www.zurich.ibm.com/~jca/papers/camvan02.pdf.
9. S. Chari, C. S.. Jutla, and A. Roy. Universally composable security analysis of oauth v2.0. *IACR Cryptology ePrint Archive*, 2011:526, 2011.
10. G. Cole. OASIS Service Provisioning Markup Language (SPML) Version 2. OASIS Standard, April 2006. http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip.
11. R. Dhamija and A. Perrig. Déja vu: A user study using images for authentication. In *Proceedings of the 9th USENIX Security Symposium*, 08 2000.
12. J. Eichholz, D. Hühnlein, and J. Schwenk. Samlizing the european citizen card. In *Proceedings of BIOSIG 2009: Biometrics and Electronic Signatures*, volume 155 of *Lecture Notes in Informatics (LNI)*, pages 105–117. GI-Edition, 2009.
13. M. Franke and O. Pfaff. Samlized kerberos. In *Sicherheit*, volume 62 of *LNI*, pages 297–308. GI, 2005.
14. S. Gajek. A universally composable framework for the analysis of browser-based security protocols. In Joonsang Baek, Feng Bao, Kefei Chen, and Xuejia Lai, editors, *Provable Security – Second International Conference, ProvSec 2008, Shanghai, China, October 30–November 1*, volume 5324 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2008.
15. S. Gajek, T. Jager, M. Manulis, and J. Schwenk. A browser-based kerberos authentication scheme. In Sushil Jajodia and Javier López, editors, *Computer Security – ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, volume 5283 of *Lecture Notes in Computer Science*, pages 115–129. Springer, August 2008.
16. T. Groß. Security Analysis of the SAML Single Sign-on Browser Artifact Profile. In *Annual Computer Security Applications Conference, December 8-12, 2003, Aladdin Resort & Casino Las Vegas, Nevada, USA*, 2003.
17. T. Groß and B. Pfitzmann. SAML artifact information flow revisited. In *In IEEE Workshop on Web Services Security (WSSS)*, pages 84–100, Berkeley, May 2006. IEEE.
18. T. Groß, B. Pfitzmann, and A.-R. Sadeghi. Browser model for security analysis of browser-based protocols. In *ESORICS: 10th European Symposium on Research in Computer Security*, volume 3679 of *Lecture Notes in Computer Science*, pages 489–508, Berlin, Germany, 9 2005. Springer-Verlag.
19. F. Hirsch, R. Philpott, and E. Maler. Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, 15.03.2005, 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf.
20. D. Hühnlein and M. Bach. How to use iso/iec 24727-3 with arbitrary smart cards. In *TrustBus 2007*, volume 4657 of *LNCS*, pages 280–289, 2007.
21. B. Ives, K. R. Walsh, and H. Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4):75–78, 2004.
22. Java Community Process. Java Authentication Service Provider Interface for Containers. Java Specification Requests (JSR) 196. http://www.jcp.org/en/jsr/detail?id=196.
23. J. Linn and P. Mishra. SSTC Response to 'Security Analysis of the SAML Single Sign-on Browser – Artifact Profile'. OASIS Working Draft 01, 24.01.2005, 2005. http://www.oasis-open.org/committees/download.php/11191/sstc-gross-sec-analysis-response-01.pdf.
24. P. G. Neumann. Risks of passwords. *Commun. ACM*, 37(4):126, 1994.
25. H.-K. Oh and S.-H. Jin. The security limitations of sso in openid. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 3, pages 1608–1611. IEEE, 2008.
26. C. Paquin. U-Prove Cryptographic Specification V1.1. Microsoft, 02 2011. http://www.microsoft.com/u-prove.
27. C. Paquin. U-Prove Technology Overview V1.1. Microsoft, 03 2011. http://www.microsoft.com/u-prove.
28. B. Pfitzmann and M. Waidner. Analysis of liberty single-sign-on with enabled clients. *Internet Computing, IEEE*, 7(6):38–44, 2003.
29. D. Reed, L. Chasen, and W. Tan. Openid identity discovery with xri and xrds. In *IDtrust '08: Proceedings of the 7th symposium on Identity and trust on the Internet*, pages 19–25. ACM, 2008.
30. J. Schwenk, L. Liao, and S. Gajek. Stronger bindings for saml assertions and saml artifacts. In *Proceedings of the 5th ACM CCS Workshop on Secure Web Services (SWS'08)*, pages 11–20. ACM Press, 2008.

31. J. Somorovsky, A. Mayer, J. Schwenk, M. Kampmann, and M. Jensen. On Breaking SAML: Be Whoever You Want to Be. Proceedings of the 21st USENIX Security Symposium, 2012.
32. P. Sovis, F. Kohlar, and J. Schwenk. Security analysis of openid. In *Proceedings of Sicherheit 2010*, Lecture Notes in Informatics (LNI). GI-Edition, 2010.
33. S.-T. Sun. Simple but not secure: An empirical security analysis of oauth 2.0-based single sign-on systems. http://blogs.ubc.ca/computersecurity/files/2012/04/San-Tsai.pdf.
34. S.-T. Sun and K. Beznosov. The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 378–390. ACM, 2012.
35. Sun Inc. Java Authentication and Authorization Service (JAAS). Reference Guide for the Java TM SE Development Kit 6. http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html.
36. H. Tschofenig and P. Hunt. Oauth 2.0 security: Going beyond bearer tokens. IETF Internet Draft (draft-tschofenig-oauth-security-01), expires 19.06.2013, 2013. http://tools.ietf.org/html/draft-tschofenig-oauth-security-01.
37. B. van Delft and M. Oostdijk. A security analysis of openid. In *Policies and Research in Identity Management*, pages 73–84. Springer, 2010.
38. M. Zhou, R. Zhang, D. Zeng, and W. Qian. Services in the cloud computing era: A survey. In *Universal Communication Symposium (IUCS), 2010 4th International*, pages 40–46. IEEE, 2010.
39. D. Fett, R. Küsters, and G. Schmitz. An Expressive Model for the Web Infrastructure: Definition and Application to the BrowserID SSO System, http://arxiv.org/abs/1403.1866.

Dr. Detlef Hühnlein
ecsec GmbH, Sudetenstraße 16, 96247 Michelau, Deutschland
detlef.huehnlein@ecsec.de

Detlef Hühnlein is CEO of ecsec GmbH (www.ecsec.de), has more than fifteen years of professional experience in the area of IT-security, received a doctoral degree in cryptography from TU Darmstadt and was lecturer at TU Vienna, TU Darmstadt and several universities of applied science for subjects like cryptography, electronic payment mechanisms, electronic signature technology, signature law, web security and last but not least identity management. He co-authored more than 60 refereed publications, has been involved in several initiatives related to electronic signatures and identity management and is active in different standardization committees within DIN, CEN, ISO and OASIS.

M. Eng. Tobias Wich
ecsec GmbH, Sudetenstraße 16, 96247 Michelau, Deutschland
tobias.wich@ecsec.de

Tobias Wich has several years of experience in the area of functional programming, language design and the development of innovative IT-solutions. He studied computer science and holds a master degree (M. Eng.) in information technology from Coburg University of Applied Sciences. He is working as consultant and software engineer at ecsec GmbH (www.ecsec.de) with a special focus on smart cards, service oriented architectures, identity management and web application security.

M. Eng. Johannes Schmölz
ecsec GmbH, Sudetenstraße 16, 96247 Michelau, Deutschland
johannes.schmoelz@ecsec.de

Johannes Schmölz has several years of experience in the area of smart cards and identity management. He studied computer science and holds a master degree (M. Eng.) in information technology from Coburg University of Applied Sciences. He gives lectures at Coburg University of Applied Sciences and is working as consultant and software engineer at ecsec GmbH (www.ecsec.de) with a special focus on smart cards, electronic signatures and identity management.

B. Sc. Hans-Martin Haase
ecsec GmbH, Sudetenstraße 16, 96247 Michelau, Deutschland
hans-martin.haase@ecsec.de

Hans-Martin Haase studied bioinformatics at the University of Jena and holds a bachelor degree (B. Sc.). After his study he joined ecsec GmbH (www.ecsec.de) and works there as consultant and software developer in the area of identity management and smart cards.