SECURITY COMMUNICATION LAYER FOR PUBLIC DISTRIBUTED REPORTING SERVICES

Decebal Popescu, Nirvana Popescu, Florin Pop*, Vlad Ciobanu, Ciprian Dobre, Valentin Cristea

University POLITEHNICA of Bucharest, Faculty of Automatic Control and Computers, Department of Computer Science
Spl. Independentei, 313, Bucharest 060042, Romania
E-mails: {decebal.popescu, nirvana.popescu, florin.pop, ciprian.dobre, valentin.cristea}@cs.pub.ro, vlad.ciobanu@cti.pub.ro

KEYWORDS

Security, Distributed Electronic Services, Electronic Identity, Communication Protocol.

ABSTRACT

The electronic management of user identities is a requirement for many modern applications. The user's identity defines its possible range of actions, and its management becomes critical in applications such as e-Banking, e-Payment, etc. Current solutions are based on methods that use certificate infrastructures, role and policy enforcement management, trust management using social networking, etc. We propose a solution for electronic management of identities that provides secure identification of a user using its electronic identity card (eID). The proposed nPA (the new German Identity card) Connector offers a trusted infrastructure for secure handling of electronic identities over the Internet. The nPA connector uses certificates obtained and guaranteed by a trusted Identity Provider. The user's personal data from the electronic Identity Card is transmitted from an original source service provider to subsequent destination service providers, all of which have previously signed a contract with the Identity Provider. The connector can be easily integrated within an application, providing a supplementary security layer for identity management. It can also be accessed remote as a Web service. In such cases the connector can be accessed by applications that can communicate with an Identity Provider from a trusted list of eID Service Providers. For that, the connector offers an interface for the application to query attributes from the electronic Identity card. The nPA connector can be considered a service provider between a user wielding a user agent (usually a web application accessed through a web browser) and an Identity Provider.

INTRODUCTION

We live in an electronic world governed by necessity to move all citizen services in electronic distributed environments. In the context e-Services, the number of users that want to move from physical world into digital world grows exponential in last decade. The e-Services are easy to use, permanent and they have continuous access, direct communication, timely and consistent information. The e-Services are exposed to all kind of threads from the Internet, so when it comes about their security or other challenges, it should receive a special attention. In a digital world, when it comes about e-Services, there are some main characteristics that are common to all of them. One of them is scalability, which indicates its ability to handle growing amounts of work in a graceful manner or its ability to be enlarged, so the distributed environments are required.

Security challenges are imposed at each step. Data encryption, password protection and account creation are other subjects discussed and applied during the development of the e-Service system. A large number of users characterize e-services and they must be able to respond to all their requests.

We propose a solution for electronic management of identities that provides secure identification of a user using its electronic identity card (eID). The proposed connector library represents a solution for the simple integration of the new German Identity card (nPA) into web applications by taking away the complexity of handling the communication with the Identity Server. The nPA connector provides the benefit of a service that offers the mechanism for sharing authentication data between trusted applications. The attributes that an application is allowed to query are specified within the contract signed with a trusted Identity Provider. The nPA Connector allows security systems and application to be developed and evolve independently.

The paper is structured as follow: Section 2 presents the related work in the field of security communication and eID. Section 3 presents the proposed architecture and in Section 4 the implementation details are presents. We present the conclusions and future work in Section 5.

RELATED WORK

Across Europe electronic identity (e-ID) card schemes are emerging. The motivation for their deployment varies from country to country, and hence also their ability to interoperate. National schemes for each country are defined by government agencies and application usage by non-government entities has been limited [1]. A very common situation is that for each service, users must remember the associated name and password they are registered under.

[*] Corresponding Author

This method is prone to identity theft and its usability leaves much to be desired. The Trusted Platform Module (TPM) proposed in [2] is a microcontroller with cryptographic functions that is integrated into many computers. Using communication services like voice services, chat services and web 2.0 technologies (wikis, blogs, etc.) are a common part of everyday life in a personal or business context. These communication services typically authenticate participants. Identities identify the communication peer to users of the service or to the service itself. Calling line identification used in the Session Initiation Protocol (SIP) can be used for Voice over IP (VoIP) [3].

In [4] is dealing with the use of the Belgian electronic ID card to secure Presence notifications in the Session Initiation Protocol (SIP). More specifically, it addresses the secure authentication to a SIP registrar server thanks to the Belgian electronic ID card. The proposed solution consists in adding an Authenticated Identity Body (AIB) to the REGISTER requests issued by the user's agent, which simultaneously authenticates the user and protects the request header.

The Italian Electronic Identity Card (EIC, for short) is a polycarbonate smart card equipped with a microchip (supporting cryptographic functions) and a laser band (featuring an embedded hologram). It contains personal (e.g. name, surname, date of birth, etc.) and biometric data (photo and fingerprint) of a citizen [5]. Currently, Republic of Turkey has introduced a new smart card based electronic identity card. By the help of this card, e-government projects of Turkey are expected to accelerate. Turkey is also prepared to use biometrics in citizen authentication. In this paper, we will shortly mention about physical and electronic properties of this card, use of biometrics, cryptographic details of the card, Electronic Authentication System and roll out of the card [6].

The new identity card (“Neuer Personalausweis”, nPA) was introduced in Germany in 2010. It supports the Federal Government's eCard strategy. The nPA is part of the nationwide introduction of the use of smart cards in the federal administration. The eCard-API-Framework is a technical frame for implementing the eCard strategy and is specified in the technical guideline BSI TR-03112 of the Federal Office for Information Security (BSI) [7]. The basic goal is to expand the conventional use of the identity card to the electronic world, thus enabling a secure and legally binding communication on the Internet [8].

NPA CONNECTOR ARCHITECTURE

The main principle for using the nPA connector resides on the usage of certificates obtained from a trusted Identity Provider that specifies the data that an application is allowed to query from the identity card when interacting with the back-end user through the citizen’s application.

The nPA Connector library represents a solution for the simple integration of nPA into web applications by taking away the complexity of handling the communication with the Identity Server.

The nPA Connector offers a trusted infrastructure for secure handling of electronic identities in the Internet for a variety of applications. The connector allows querying the attributes from the electronic Identity card providing also a useful tool for securely validating information about the citizens. The connector makes it easy for web applications to communicate with an Identity Provider from the eID Service Providers trusted list because it is mainly a mechanism for sharing authentication data between trusted applications (see Figure 1). The user’s personal data from the electronic Identity Card is transmitted from an original source service provider to subsequent destination service providers, all of which have previously signed a contract with the Identity Provider, allowing them to query particular information about the user, and in agreement with the citizen’s options for sharing their personal data.

At a deeper level, the nPA connector acts like a Service provider between a user wielding a user agent (usually a web application accessed through a web browser) and an Identity Provider. The user requests a web resource from the application which is in fact protected by the Service provider responsible for creating a secure context for querying sensitive data.

The service provider, who wishes to know the identity of the requesting user, and also personal information about the user, issued the authentication request to the Identity Provider through the user agent.

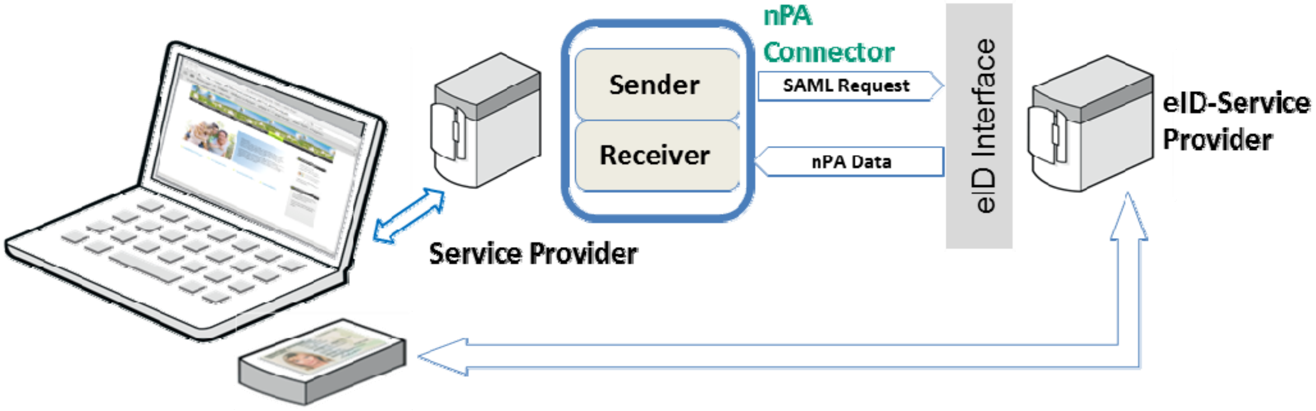

Figure 1. nPA Architecture and interactions

IMPLEMENTATION DETAILS

This library allows security systems and application software to be developed and evolve independently, and also allows decoupling of the application software from the underlying security infrastructure. This is because SAML provides a set of interoperable standard interfaces. Standardizing the interfaces between systems allows for faster, cheaper, and more reliable integration. Furthermore, this library provides the possibility to configure custom extensions of the profiles of SAML usage, and the benefits that come from this customization open up more and different kinds of access management.

Following are some more concrete benefits of this connector brought by the usage of SAML protocol:

- *Platform neutrality*. Abstractization of the security framework away from platform architectures and particular vendor implementations. Making security more independent of application logic is an important tenet of Service-Oriented Architecture.

- *Improved online experience for end users*. It enables single sign-on by allowing users to authenticate at an identity provider and then access service providers without additional authentication by creating security contexts.

- *Reduced administrative costs for service providers*. The burden of maintaining account information burden is transferred to the identity provider.

- *Risk transference*. It pushes responsibility for proper management of identities to the identity provider, which is more often compatible with its business model than that of a service provider.

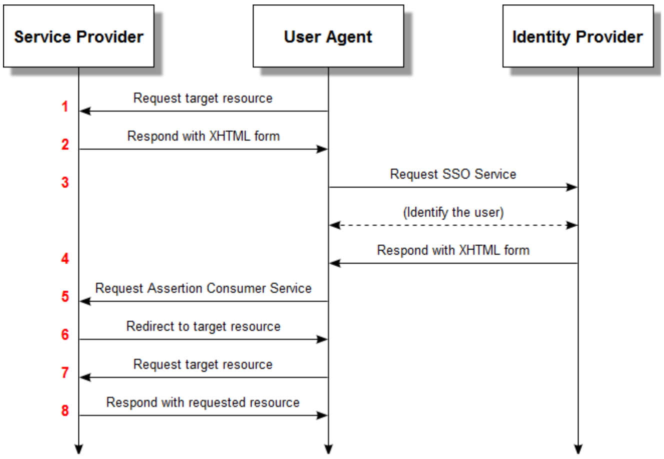

Figure 2. nPA protocol

The main functionality of the nPA connector library is based on the SAML 2.0 protocol [9]. SAML is a XML-based protocol whose functionality is defined by four basic concepts: assertions, protocols, bindings and profiles. Assertions are XML-based messages that are formed using the rules of syntax and semantics defined in the SAML Core. These assertions are requested and transmitted using one of the specified protocols from one system entity to another.

The requested attributes and equivalent response attributes are defined in a custom schema, specified by the identity Provider server and imply custom made Marshallese and Un-Marshallese to build objects from and to XML files. There are two major types of configurations for profiles: profiles for requesting a number of specific attributes of the user for the Identity Provider; profiles for verifying the value of certain key attributes against a given value.

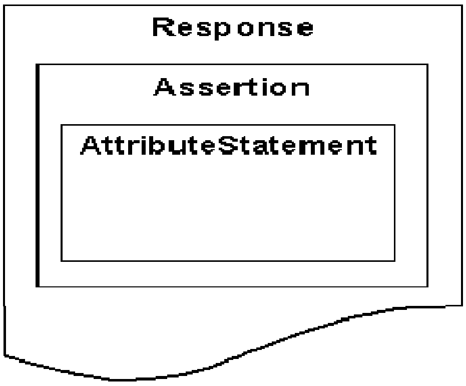

Figure 3. SAML message

The main functionality of the protocol provided by this library is illustrated in Figure 2, where there nPA connector (Service Provider) is responsible for creating a security context for the user agent. The nPA Connector uses the HTTPPost binding from SAML protocol to communicate with the user agent (see Figure 3).

The message flow of the protocol begins with a request of a secured resource at the Service Provider and follows these steps (presented in Figure 2):

1. Request a target resource at the SP

- For example https://sp.example.com/myresource
- The library performs a security check on behalf of the target resource : if a valid security context already exists skip steps 2 – 7

2. Respond with an XHTML form

```
<form method="post"
action="https://idp.example.org/SAML2/SSO/
POST" ...>
 <input type="hidden" name="SAMLRequest"
value="request" />
   <input type="hidden" name="RelayState"
value="token" />
   ...
   <input type="submit" value="Submit" />
 </form>
```

- The value of the SAMLRequest parameter is the base64 encoding of the actual <samlp:AuthnRequest> element

3. Request the SSO Service at the IdP

- The user agent issues a POST request to the SSO service at the identity provider

```
POST /SAML2/SSO/POST HTTP/1.1
Host: idp.example.org
Content-Type:
application/x-www-form-urlencoded
Content-Length: nnn
SAMLRequest=request&RelayState=token
```

4. Respond with an XHTML form

- The SSO service validates the request and responds with a document containing an XHTML form:

```
<form method="post"
action="https://sp.example.com/SAML2/SSO/POST
" ...>

<input type="hidden" name="SAMLResponse"
value="response" />

<input type="hidden" name="RelayState"
value="token" />

   ...

<input type="submit" value="Submit" />
</form>
```

- The value of the SAMLResponse parameter is the base64 encoding of the actual <samlp:Response> element.

5. Request the Assertion Consumer Service at the SP

```
POST /SAML2/SSO/POST HTTP/1.1
Host: sp.example.com
Content-Type:
application/x-www-form-urlencoded
Content-Length: nnn
SAMLResponse=response&RelayState=token
```

6. Redirect to target resource

- The library creates a security context and redirects the user to the request resource

7. Request the target resource at the SP again

8. Respond with the requested resource

Since a security context exists, the service provider returns the resource to the user agent.

The requested attributes, are encrypted using a symmetric key and sent as a protocol message using the HTTP POST binding. The SSO service at the Identity provider validates the request and responds with a document containing the response. The value of the SAMLResponse parameter is the base64 encoding of a <samlp:Response> element, which likewise is transmitted to the service provider via the browser.

The nPA Connector stands for an extensible and configurable plugin for web based applications that wish to communicate with an Identity Provider with which they have previously signed an agreement and obtained a certificate that allows them to query certain sensitive information about the citizens. The configuration of the connector can easily be done by altering the configuration file (as seen in the examples in previous chapter).

For example, as use case, if an user has signed on and authenticated at a portal like http://www.rentamovie-exampleportal.com and wish to rent a movie, the website will only get access to the attributes specified within the contract of that website (for example the age of the end-user) with the Identity Provider, and in agreement with the user's preferences for displaying sensitive data (the user may not agree to share other information).

At a highlevel, the entry point in the library is represented by the `eIDHandler` which is an abstraction for handling both the request and the response messages as illustrated in Figure 4.

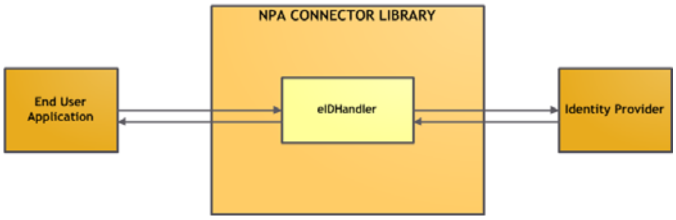

Figure 4. Entry point for nPA

The `eIDHandler` is actually an abstraction layer that offers methods like `handleRequest()`, `handleResponse()` and in fact uses dedicated handlers for request (`eIDRequestHandler` – responsible for creating custom extensions for the given profile), and response (`eIDResponseHandler` – responsible for reading the custom extensions from the response assertion received from the identity provider server). The core functionality of the SAML protocol is described and implemented in third party libraries and do not make the object of this document.

The message flow of the protocol begins with a request of a secured resource at the Service Provider and follows these steps: request a target resource at the SP, respond with an XHTML form, request the SSO Service at the IdP, respond with an XHTML form, request the Assertion Consumer Service at the SP, redirect to target resource, request the target resource at the SP again, respond with the requested resource. The nPA library uses the Authentication Request Protocol, as specified in SAML Core [9] to communicate with the Identity Provider. The overall flow of the authentication process through the library is shown in the Figure 5.

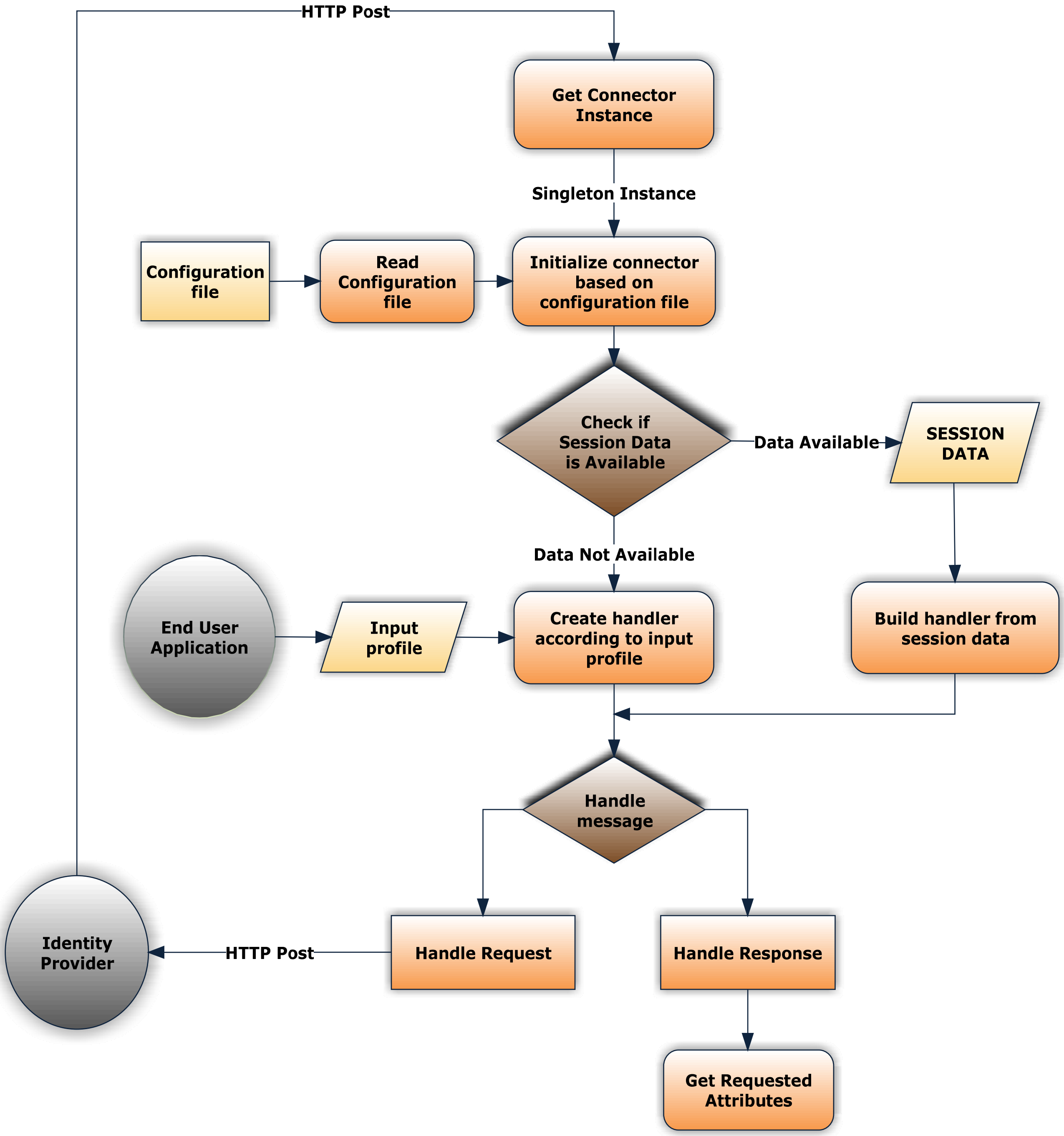

Figure 5. nPA authentication process

CONCLUSIONS

We present in this paper the nPA connector architecture and implementation details. The nPA Connector consists in an application which allows a secure identification of a certain user using an electronic identity card (eID) over the Internet.

The nPA Connector is designed for PHP and JAVA implementation using the Apache Server and will be available under different OS, both Windows and Linux environments. The nPA Connector was tested on Windows 2008 and using IIS7. The solution uses the SAML library which has a MPL license.

ACKNOWLEDGMENTS

The research presented in this paper is supported by national project: "SORMSYS - Resource Management Optimization in Self-Organizing Large Scale Distributes Systems", Contract No. 5/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 201. The work has been co-funded by the Sectorial Operational Program Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557. The research presented in this paper was implemented in the international project called “PrO”, developed by Centre for Advanced Studies on Electronic Services (e-CAESAR).

REFERENCES

Siddhartha Arora. 2008. National e-ID card schemes: A European overview. *Inf. Secur. Tech. Rep.* 13, 2 (May 2008), 46-53.

Andreas Klenk, Holger Kinkelin, Christoph Eunicke, and Georg Carle. 2009. Preventing identity theft with electronic identity cards and the trusted platform module. In *Proceedings of the Second European Workshop on System Security* (EUROSEC '09). ACM, New York, NY, USA, 44-51.

Rainer Falk, Steffen Fries, and Hans Joachim Hof. 2010. Protecting Voice over IP Communication Using Electronic Identity Cards. In *Proceedings of the 2010 Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services* (CENTRIC '10). IEEE Computer Society, Washington, DC, USA, 5-10.

Sebastien Gamby, Laurent Schumacher, and Jean Ramaekers. 2007. Securisation of SIP Presence notifications thanks to the Belgian electronic identity card. In *Proceedings of the The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies* (NGMAST '07). IEEE Computer Society, Washington, DC, USA, 125-129.

Franco Arcieri, Mario Ciclosi, Fabio Fioravanti, Enrico Nardelli, and Maurizio Talamo. 2004. The Italian electronic identity card: a short introduction. In *Proceedings of the 2004 annual national conference on Digital government research* (dg.o '04). Digital Government Society of North America , Article No: 73.

Mucahit Mutlugun and Oktay Adalier. 2009. Turkish national electronic identity card. In *Proc. of the 2nd int. conf. on Security of information and networks* (SIN '09). ACM, New York, NY, USA, 14-18.

BSI: Advanced Security Mechanisms for Machine Readable Travel Documents; Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI); Version 2.03. Technische Richtlinie TR-03110, 2010.

Margraf, Marian: *Der elektronische Identitätsnachweis des zukünftigen Personalausweises*. SIT-SmartCard Workshop 2009, Darmstadt, 2009.

Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, and Llanos Tobarra. 2008. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for Google Apps. In *Proceedings of the 6th ACM workshop on Formal methods in security engineering* (FMSE '08). ACM, New York, NY, USA, 1-10.