SAMLizing the European Citizen Card

(Extended Abstract)*

Jan Eichholz[1], Detlef Hühnlein[2], Jörg Schwenk[3]

[1] Giesecke & Devrient GmbH, Prinzregentenstraße 159, 81677 München,
jan.eichholz@gi-de.com

[2] secunet Security Networks AG, Sudetenstraße 16, 96247 Michelau,
detlef.huehnlein@secunet.com

[3] Ruhr Universität Bochum, Universitätsstr. 150, 44780 Bochum
joerg.schwenk@rub.de

Abstract: While the use of Federated Identity Management and Single Sign-On based on the Security Assertion Markup Language (SAML) standards becomes more and more important, there are quite a few European countries which are about to introduce national ID cards, which are compliant to the European Citizen Card (ECC) specification prTS 15480. The present contribution shows how these two seemingly opposite approaches may be integrated in a seamless and secure fashion such that it is possible to use the security features of the ECC in a federated scenario, which allows easy integration of Service Providers.

1 Introduction

In the area of Identity Management there seem to be two major trends at the moment, which are addressed in the EU funded project STORK[1]: On the one hand side, Federated Identity Management solutions are increasingly used in practice as they allow to implement Single Sign-On and facilitate the integration of Service Providers. The Security Assertion Markup Language (SAML), which has been developed by OASIS, plays a central role in the implementation of Federated Identity Management. On the other side quite a few European countries are about to introduce national ID cards, which are compliant to the European Citizen Card specification [CEN15480-1, CEN15480-2, CEN15480-3, CEN15480-4]. Hence it is natural to investigate how both approaches can be integrated such that systems which aim at implementing the eService directive [2006/123/EC] may combine the security of the ECC with the easy integration of Service Providers in SAML.

The rest of the paper is structured as follows: Section 2 explains how the ECC may be "SAMLized" and how an ECC-specific SAML-profile may look like, which may be used as starting point for the development of further specifications in STORK and standardization in CEN TC 224 WG 15 and/or OASIS Security TC. Within the full paper, a

*The full paper is available at http://www.ecsec.de/pub/SAMLizing-ECC.pdf.

[1] See www.eid-stork.eu

background chapter provides the necessary information on the Security Assertion Markup language and the European Citizen Card supporting the Extended Access Control (EAC) protocol [BSI-TR-03110(V2.01)] and briefly considers related work.

2 Secure Integration of the ECC into a SAML-environment

In order to implement the eService-Directive [2006/123/EC] it is necessary that European citizen are able to use their national identification token (e.g. an ECC-compliant ID-card) to authenticate at some eService in another EU Member State. As long as not all eServices across Europe directly support the ECC-compliant authentication protocols, such as EAC for example, the use of Federated Identity Management techniques, e.g. based on SAML, may ease the integration of eServices and hence facilitate the implementation of the eService-Directive. On the other side it is necessary to seriously analyze security aspects of such a construction, as a naive integration of a highly secure national ID-card into a SAML-environment may considerably degrade the overall security.

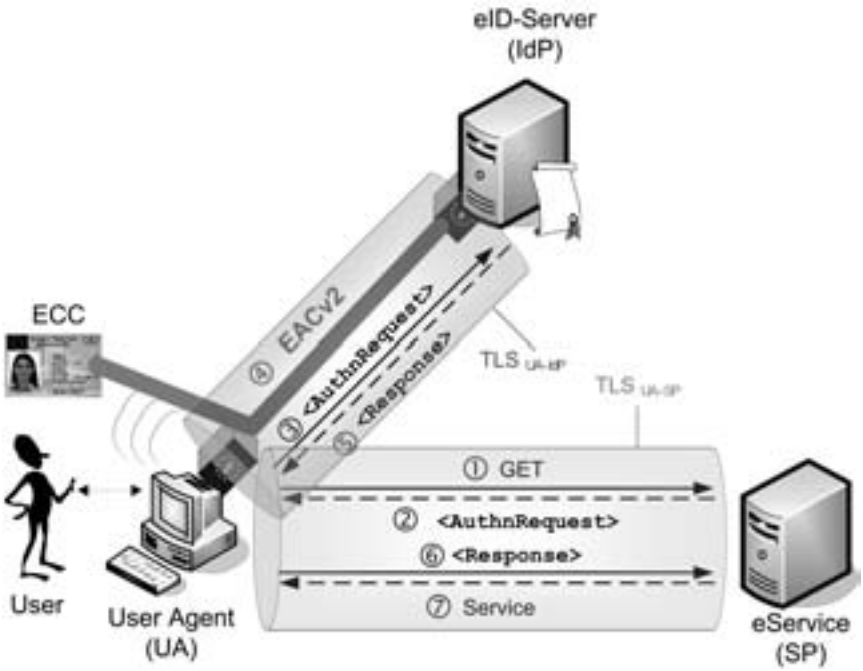

Figure 1: Combined ECC-3 and SAML architecture

As explained in Section 2.1 there are three main approaches for the secure integration of the European Citizen Card into a SAML-environment. First SAML may be bound to the involved TLS-sessions (cf. Section 2.2). Second the two TLS-sessions may be bound together and may be bound to the EAC-session (cf. Section 2.3). Third it is possible to

bind the SAML-Assertion directly to the EAC-protocol (cf. Section 2.4). Finally, the pros and cons as well as the possible combination of these approaches are discussed in Section 2.5.

2.1 Overview, requirements and threats

In order to allow Users, which are equipped with EAC-based eID tokens, to access the services of a Service Provider (SP), which only supports SAML, it is a straightforward approach to make use of a specific eID-Server, which supports both EAC and SAML and may serve as Identity Provider (IdP), which "translates" an EAC-based authentication context into an appropriate SAML-Assertion, which may be consumed by the Service Provider.

The applied protocol (see Figure 1) is very similar to the SAML-protocol for an enhanced client, which is capable of performing an EAC-based authentication in step 4. Furthermore it can be seen in Figure 1 that besides the EAC-channel between the ECC and the eID-Server there may be two TLS-channels (TLS_{UA-SP} between the User Agent and the Service Provider (eService) and TLS_{UA-IdP} between the User Agent and the Identity Provider (eID-Server)).

The major goal for the integration of the ECC into a SAML-environment is that the Service shall be accessible to the User (Agent) in step (7) if and only if an EAC-based authentication has been successfully performed in step (4). Furthermore it may be desirable to have the option to include cryptographic evidence into the SAML-Assertion transported in steps (5) and (6) such that it can be proved at a later point in time (e.g. at court) that the SAML-Assertion indeed was generated with a valid European Citizen Card (ECC).

However as explained in [SAML-SecP(v2.0)] there are a number of threats against SAML-based solutions, which need to be considered to end up with a secure system. We only consider man-in-the-middle (MitM) attacks and refer to [SAML-SecP(v2.0)] for other security aspects related to SAML.

If the TLS-channels are established in an anonymous mode, in which no X.509-certificates are used, it is clear that an attacker may mount a MitM-attack as depicted in Figure 2, steal the SAML-Assertion contain in the `Response`-element in order to impersonate the User at the eService.

In a similar fashion an attacker may mount a MitM-attack, if only the TLS-servers (i.e. the eID-Server, the eService and the attacker) are equipped with X.509-certificates and the User is not able to recognize the difference between the certificates presented within the TLS-handshakes. Note that this is a realistic assumption since studies have shown that typical internet users tend to ignore TLS security indicators [DTH06], and that it currently may even be possible to fake trustworthy looking TLS server certificates [SLW09].

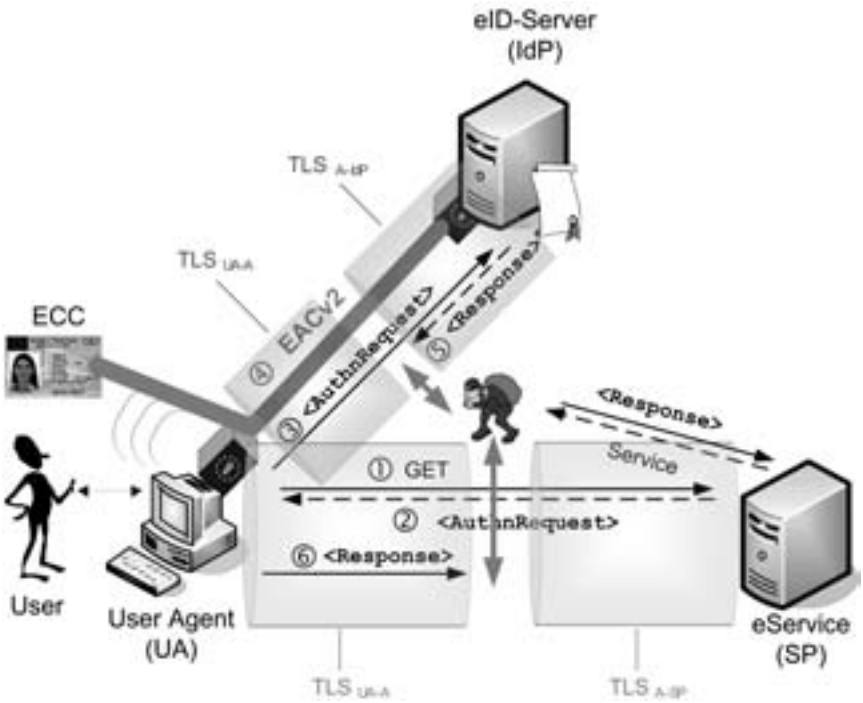

Figure 2: Man-in-the-Middle-Attack against SAML

2.2 Secure Binding of SAML to TLS

In order to thwart attackers, which try to steal a SAML token, e.g. Assertion or an Artifact, one may provide a cryptographic binding between the SAML token and the underlying TLS-channel.

In previous work, we identified three methods to bind SAML tokens to a specific TLS session. By binding the token to the session, the eService may deduce that the data he sends in response to the SAML token will be protected by the same TLS-channel, and will thus reach the same client who has previously sent the token.

- **TLS Federation [BHS08].** In this approach, the SAML token is sent inside an X.509 client certificate. The SAML token thus may replace other identification data like distinguished names. The certificate has the same validity period as the SAML token.
- **SAML 2.0 Holder-of-Key Web Browser SSO and Assertion Profile [SAML-HoKAP, SAML-HoKWebSSO].** Here again TLS with client authentication is used, but the client certificate does not transport any autorization information. Instead, the SAML token is bound to the public key contained in this certificate, by including this key in a Holder-of-Key assertion. The security of this approach has

independently been analyzed in [GJMS08].

- **Strong Locked Same Origin Policy [GLS2008].** Whereas the previous approaches relied on the server authenticating (in an anonymous fashion) the client, in this approach we strengthen the client to make reliable security decisions. This is done by using the servers public key as a basis for decisions of the Same Origin Policy, rather than the insecure Domain Name System.

2.3 Secure Binding of TLS to EAC

Since the EAC authentication can be performed over any communication link, it is even possible to successfully complete it over two TLS-channels between the User Agent and the eID-Service with a MitM-attacker in between (cf. Figure 2). Note that the MitM-attack does not affect the EAC-authentication itself, but only allows the attacker to intercept the SAML-Assertion, which is issued as a result of the EAC-authentication. In order to avoid this kind of attack one may include TLS-specific values in the EAC-protocol in order to provide a cryptographic binding between TLS and EAC.

For this purpose we first investigate *which* TLS-specific parameters may be included into the EAC protocol and then we briefly discuss *how* these values may precisely be incorporated into EAC such that the TLS- and EAC-channels are cryptographically tied together.

2.3.1 TLS-specific parameters for potential inclusion in EAC

We consider the following values from a TLS handshake for inclusion in EAC:

- **Certificates or other messages of the TLS-Handshake Protocol.** While it should be easy to access these values using a browser plugin or a server component, it would not be sufficient to use those parameters as they do not depend on both communication partners. Furthermore the used certificates are typically not session specific.
- **Premaster secret.** This value can only be used if a cipher suite using Diffie-Hellman key exchange is chosen. If RSA encryption is used, the MitM-attacker can simply decrypt the premaster secret chosen by the browser, and re-encrypt it for the server.
- **Master secret.** The master secret, or any value derived from it, can be used, since the two nonces sent by browser and server are used to compute it. While a derivation mechanism for the master secret is described in [Resc09] this mechanism does not seem to be supported by popular browsers.
- **Finished message.** Another approach would be to use one of the two Finished messages, since this value is derived from the master secret, and it is sent protected only by the TLS record layer. Thus it should be easy for a browser plugin, or a server component, to access it.

- **Pre-shared key between the eID-Server and eService.** In [BSI-TR-03112-7] it is described how to provide a binding between the two TLS-connections using a pre-shared-key (PSK) known to the eID-Server and the eService. The PSK may be generated by the eID-Server, the eService or both and is transported from the eService to the User Agent over the first TLS-channel (TLS_{UA-SP} in Figure 1) and used for the establishment of the second TLS-channel between the User Agent and the eID-Server (TLS_{UA-IdP} in Figure 1) as specified in [RFC4279].

  In addition to the binding of the two TLS-channels the PSK may also be used to provide a binding of the TLS-channels to the EAC-channel.

In particular the last two values seem to fulfill our requirements very well and may serve as input for a binding of TLS to EAC.

2.3.2 Integration of TLS-specific values into EAC

It remains to discuss how the TLS-specific values may be integrated into EAC. For this purpose there are the following general options induced by the structure of the EAC-protocol:

- **Terminal Authentication.** The Terminal Authentication (cf. [BSI-TR-03110(V2.01), Section 4.4]) roughly consists of the following three steps:

  1. As a first step in the Terminal Authentication protocol the ECC verifies the Card-verifiable-Certificate (CVC) provided by eID-Server.

     In order to provide a cryptographic link between the X.509 certificate used in TLS and the CVC used in EAC it would be possible to include (a hash value of) one certificate as an extension into the other certificate. For the inclusion of the CVC into an X.509-certificate one may use the `CVCert`-extension defined in [ISO18013-3, Section C.7.2.1]. In order to include the hash value of an X.509-certificate in a CVC it would be necessary to define a corresponding extension in an amendment of [BSI-TR-03110(V2.01), Annex C.3]. On the other side it would – from a theoretical point of view – be possible that the Card-verifiable-Certificates are directly used in TLS in a similar fashion as one may use OpenPGP-keys (cf. [RFC5081]).

  2. Next the eID-Server generates an ephemeral key pair, which is especially used in the Chip Authentication protocol described below. As explained in Section 2.4 the private ephemeral key may be derived from a secret, which is shared by the eService and the eID-Server.

  3. Finally a challenge is obtained from the ECC and signed by the eID-Server.

     This challenge contains an identifier derived from the ephemeral PACE-key of the ECC, a nonce generated by the ECC, an identifier derived from the ephemeral public key of the eID-Server which is generated in the previous step and possibly additional so called ”Authenticated Auxiliary Data” (AAD) (cf. [BSI-TR-03110(V2.01), Annex A.6.5]). The AAD are normally used for age verification, document validity verification and community ID verification,

but it seems to be possible to use the AAD to convey the TLS-specific value discussed above such that the TLS-channel is cryptographically bound to the EAC-channel, which effectively removes the MitM-attack described above (cf. Figure 2).

- **Chip Authentication.** In the Chip Authentication (cf. [BSI-TR-03110(V2.01), Section 4.3]) the static public key of the ECC and the ephemeral public key of the eID-Server generated in step 2 above is used to agree on a common key, which is used to derive secure messaging keys and authenticate the chip of the ECC. Without significant changing the protocol and the related smart card implementation it seems to be the only option to use the TLS-specific value as seed for the generation of the ephemeral private key of the eID-Server and the keys necessary to verify this construction would provide access to the secure messaging channel between the eID-Server and the ECC. Please refer to Section 2.4 for the use of this feature in the context of SAML.

2.4 Secure Binding of SAML to EAC

For sensitive use cases it may be necessary to enable the eService, which only has access to the SAML-Assertion, to verify that the authentication indeed has been performed using an authentic ECC and that the attributes conveyed in the SAML-Assertion indeed have been read out from the ECC in a secure EAC-session. In order to achieve this a cryptographic binding between SAML and EAC may be constructed as explained in the following.

The authentication of the ECC is achieved by the chip authentication protocol, which basically is a Diffie-Hellman (DH) key exchange using static keys on the chip side. The resulting keys are used for secure messaging later on. On the other side the eID-Server would usually generate an ephemeral DH key pair using a random seed. In our case however the ephemeral private key is derived from a shared key which has been agreed upon by the eService and the eID-Server. This allows the eService to add own random data to the key generation process and more importantly it allows the eService to verify that the authentication has been performed with a trustable ECC and that sensitive attributes contained in the SAML-Assertion indeed have been read out from the ECC in a secure EAC-session (see Figure 3).

Before sending the SAML `<AuthnRequest>` to the eID-Server, the eService generates an ephemeral DH key pair $(\tilde{S}K_{SP}, \tilde{P}K_{SP})$ and sends the public key $\tilde{P}K_{SP}$ together with the domain parameters $\mathcal{D}$ within the SAML `<AuthnRequest>` to the eID-Server. The additional data may be placed within the `<AuthnRequest>.<RequestedAuthnContext>.<AuthnMethod>.<AsymmetricKeyAgreement>` structure for example.

Upon receiving the `<AuthnRequest>` the eID-Server also generates an ephemeral DH key pair $(\tilde{S}K_{IdP}, \tilde{P}K_{IdP})$ using the domain parameters $\mathcal{D}$ chosen by the eService. Using $\tilde{P}K_{SP}$ and $\tilde{S}K_{IdP}$, the eID-Server calculates the common key which is used to derive the ephemeral private key $\tilde{S}K_{CA}$ and the corresponding $\tilde{P}K_{CA}$, which is used in the Chip Authentication protocol.

After the eID-Server has successfully performed the EAC protocol with the ECC, he received the data $(\mathcal{D}_{ECC}, PK_{ECC}, EF.CardSecurity, r_{ECC}, T_{ECC})$ from the ECC, which can be used to verify the genuineness of the ECC.

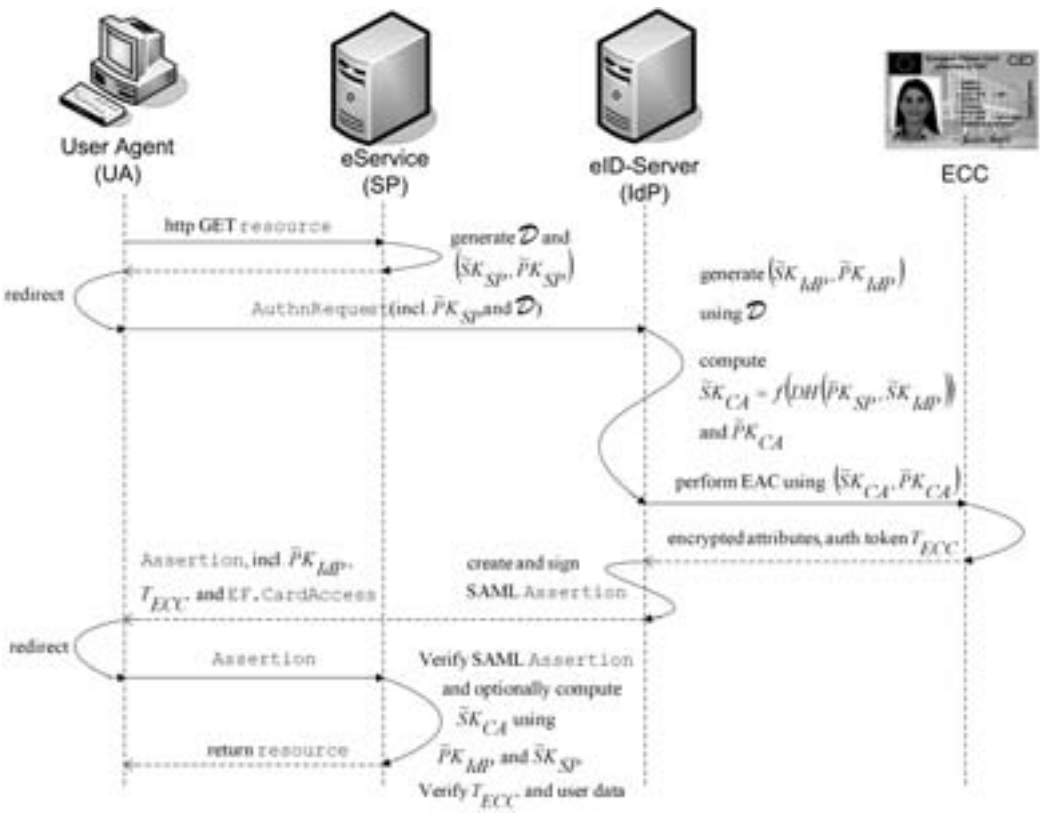

Figure 3: Command Flow for eService Authentication Token verification

Within the SAML response element `<Response>.<Assertion>.<AuthnStatement>.<AuthnContext>` – which is an instance of [SAML-Auth(v2.0), Section 3.4.15] – the necessary verification data can be placed and hence be made available to the eService. Using $\tilde{P}K_{IdP}$ and $\tilde{S}K_{SP}$ the eService is able to compute the private key $\tilde{S}K_{CA}$ used in the Chip Authentication protocol. Afterwards it may use $\tilde{S}K_{CA}$ and PK_{ECC} to compute the secure messaging keys and hence verify the validity of the authentication token T_{ECC}.

Since the eService now has the secure messaging keys of the EAC-channel, it may be possible that the eID-Server does not decrypt the data received from the ECC, but instead places the secure messaging cryptograms received from the ECC within an `Encrypted-`

`Attribute`-element within the Assertion. To retrieve the plain value of the attributes, the eService needs to decrypt the `EncryptedAttribute`-element with the derived secure messaging key.

2.5 Discussion of different approaches and recommendations

In this section it remains to discuss the different approaches presented above and derive recommendations for the secure integration of the European Citizen Card into a SAML-environment.

Since for example the mechanisms [SAML-HoKWebSSO, SAML-HoKAP] mentioned in Section 2.2 are independent from the applied authentication protocol they may of course be used in the ECC-context.

In case of an ECC which supports the EAC protocol however it is possible to provide a tighter and probably more secure binding between EAC, TLS and SAML.

Among the different options discussed in Section 2.3 one may in particular include TLS-specific values as additional "Authenticated Auxiliary Data" (AAD) into the Terminal Authentication step within the EAC protocol in order to provide a strong binding between TLS and EAC. If there is already a pre-shared key (PSK) between the eID-Server, the eService and the User Agent as required by [BSI-TR-03112-7] one may include (the hash value of) this value as AAD in EAC. Alternatively one may use the (hash value of the concatenation of the) Finished Messages of the TLS-channels as input to the EAC-protocol. While the eID-Server has direct access to the Finished Messages of TLS_{UA-IdP} the corresponding value for TLS_{UA-IdP} would need to be transported in encrypted form to the eID-Server and may be included in the optional `<Extensions>`-element within `<AuthnRequest>`.

Whether it makes sense to introduce a cryptographic link between the CVC used for EAC and the X.509-certificates used for TLS mainly depends on organizational aspects such as the respective certificate lifetime and involved enrollment procedures.

In order to provide a direct binding between SAML and EAC and especially if the eService requires a proof that the authentication was performed with a trustable ECC, it is highly recommendable to use the mechanism introduced in Section 2.4. This proposal seems to be especially attractive from a practical point of view as it may help to solve liability issues introduced by the delegation of the sensitive authentication step to the eID-Server.

Finally for maximum security one may combine the different proposals and link SAML to TLS (cf. Section 2.2), TLS to EAC (cf. Section 2.3) and SAML to EAC (cf. Section 2.4).

3 Conclusion

Based on the discussion in the previous sections it seems that the integration of the European Citizen Card into a SAML-environment has the potential to solve many open issues related to the acceptance of ECC based authentication protocols, fast deployment and easy integration into existing web service infrastructures, which already (are about to) use SAML.

However, the slightly increased complexity of the system introduces additional threats as an attacker may for example act as Man-in-the-Middle and steal the SAML-Assertion and finally impersonate the User which has been securely authenticated based on the ECC. In order to prevent such attacks various mechanisms have been proposed which provide a cryptographic binding between SAML, TLS and EAC. Furthermore the binding between SAML and EAC may be helpful to solve liability issues due to the introduction of the eID-Server acting as trusted third party.

To sum up we solved security problems which are also present in many other Federated Identity Management scenarios, we greatly simplify the introduction of ECC into existing web service infrastructures, and we introduced an approach which may help to solve liability issues related to the delegation of the sensitive authentication step.

References

[2006/123/EC] *Directive 2006/123/EC of the European Parliament and the Council of 12 December 2006 on Services in the Internal Market*. Official Journal of the European Union, L 376/36, 27.12.2006. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:376:0036:0068:EN:PDF, 2006.

[BHS08] BUD P. BRUEGGER, DETLEF HÜHNLEIN, and JÖRG 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). http://www.ecsec.de/pub/TLS-Federation.pdf.

[BSI-TR-03110(V2.01)] FEDERAL OFFICE FOR INFORMATION SECURITY (BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK). *Advanced Security Mechanism for Machine Readable Travel Documents - Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI)*. Technical Directive (BSI-TR-03110), Version 2.01. http://www.bsi.bund.de/english/publications/techguidelines/tr03110/TR-03110_v201.pdf, 2009.

[BSI-TR-03112-7] FEDERAL OFFICE FOR INFORMATION SECURITY (BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK). *eCard-API-Framework – Protocols*. Technical Directive (BSI-TR-03112), Version 1.1, Part 7. http://www.bsi.bund.de/literat/tr/tr03112/, 2009.

[CEN15480-1] COMITÉ EUROPÉEN DE NORMALISATION (CEN). *Identification card systems - European Citizen Card - Part 1: Physical, electrical and transport protocol characteristics*. CEN/TS 15480-1 (Technical Specification), 2007.

[CEN15480-2] COMITÉ EUROPÉEN DE NORMALISATION (CEN). *Identification card systems - European Citizen Card - Part 2: Logical data structures and card services*. CEN/TS 15480-2 (Technical Specification), 2007.

[CEN15480-3] COMITÉ EUROPÉEN DE NORMALISATION (CEN). *Identification card systems - European Citizen Card - Part 3: European Citizen Card Interoperability using an application interface*. CEN 15480-3 (Working Draft), 2009.

[CEN15480-4] COMITÉ EUROPÉEN DE NORMALISATION (CEN). *Identification card systems - European Citizen Card - Part 4: Recommendations for European Citizen Card issuance, operation and use*. CEN 15480-4 (Working Draft), 2009.

[DTH06] RACHNA DHAMIJA, J. D. TYGAR, and MARTI HEARST. *Why phishing works*. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590 (ACM, 2006). http://graphics8.nytimes.com/images/blogs/freakonomics/pdf/Why_Phishing_Works-1.pdf.

[GJMS08] SEBASTIAN GAJEK, TIBOR JAGER, MARK MANULIS, and JÖRG 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, 2008).

[GLS2008] JÖRG SCHWENK, LIJUN LIAO, and SEBASTAN 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).

[ISO18013-3] *ISO/IEC 18013-1: Personal Identification – ISO Compliant Driving Licence – Part 3: Access control, authentication and integrity validation*. International Standard, 2009.

[Resc09] E. RESCORLA. *Keying Material Exporters for Transport Layer Security (TLS)*. IETF Internet Draft, v6. http://www.ietf.org/id/draft-ietf-tls-extractor-06.txt, July 2009.

[RFC4279] P. ERONEN and H. TSCHOFENIG. *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*. Request For Comments – RFC 4279. http://www.ietf.org/rfc/rfc4279.txt, December 2005.

[RFC5081] N. MAVROGIANNOPOULOS. *Using OpenPGP Keys For Transport Layer Security Authentication*. Request For Comments – RFC 5081. http://www.ietf.org/rfc/rfc5081.txt, November 2007.

[SAML-Auth(v2.0)] JOHN KEMP, SCOTT CANTOR, PRATEEK MISHRA, ROB PHILPOTT, and EVE MALER. *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf, 2005.

[SAML-HoKAP] TOM SCAVO. *SAML V2.0 Holder-of-Key Assertion Profile*. OASIS Committee Draft 02, 05.07.2009. http://www.oasis-open.org/committees/download.php/33236/sstc-saml2-holder-of-key-cd-02.pdf, 2009.

[SAML-HoKWebSSO] N. KLINGENSTEIN. *SAML V2.0 Holder-of-Key Web Browser SSO Profile*. OASIS Committee Draft 02, 05.07.2009. http://www.oasis-open.org/committees/download.php/33239/sstc-saml-holder-of-key-browser-sso-cd-02.pdf, 2009.

[SAML-SecP(v2.0)] FREDERICK HIRSCH, ROB PHILPOTT, and EVE MALER. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf, 2005.

[SLW09] MARC STEVENS, ARJEN LENSTRA, and BENNE DE WEGER. *Chosen-prefix Collisions for MD5 and Applications*. Submitted to Journal of Cryptology. https://documents.epfl.ch/users/l/le/lenstra/public/papers/lat.pdf, June 2009.