eID Online Authentication Network Threat Model, Attacks and Implications

Christian J. Dietrich[1,2], Christian Rossow[1,3], and Norbert Pohlmann[1]

[1]Institute for Internet Security, University of Applied Sciences Gelsenkirchen, Germany
[2]Department of Computer Science, Friedrich-Alexander University, Erlangen, Germany
[3]Computer Systems Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Zusammenfassung

We present a threat model for network-based attacks on the eID Online Authentication. Using this model, we show why network-level man-in-the-middle attacks fail, in theory. However, practical limitations of the current eID client used with the German eID card allow an attacker to perform man-in-the-middle attacks when integrity checks of the TLS peer certificates are not applied to all connections of the corresponding channels.

1 Introduction

Since November 2010, the new electronic German ID card provides a facility to perform an online remote authentication of the ID card holder. This method is called eID Online Authentication and is defined in the Technical Guideline TR-03110 [3] of the Federal Office for Information Security (BSI). As part of the eID Online Authentication, personal data can be transmitted from the electronic ID card to its counterpart, the eID server. All data transmitted between the eletronic ID card and the eID server is supposed to be subject to secure messaging.

We develop a threat model and address the feasibility of network-level man-in-the-middle attacks against the eID Online Authentication functionality of the new German electronic ID card. Furthermore, we perform a number of man-in-the-middle attacks against the most-widely used eCard API client implementation for the eID Service, called AusweisApp. As personal data is increasingly valuable nowadays, we impersonate an attacker trying to intercept personal data that is transmitted as part of the eID Online Authentication.

The main contributions of this work are:

- We propose a threat model focussed on the network communication in the context of the eID Online Authentication.

- Based on this threat model, we describe the prerequisites of man-in-the-middle attacks and develop the required tools to perform such attacks.
- We provide the results of performing MITM attacks against the most-widely used combination of eID client application (AusweisApp) and eID server.
- We provide a free proof-of-concept implementation of TLS-RSA-PSK ciphersuites for OpenSSL [1].

This paper is structured as follows. In Section 2, we will give an overview of our threat model. Section 3 describes the prerequisites for man-in-the-middle attacks against several communication channels of the eID Online Authentication. We will then provide the results of attacks that we performed against the most-widely used combination of eID components in Section 4. Based on the attack results, we suggest practical implications in Section 5. Finally, related work is discussed in Section 6 and we provide a conclusion in Section 7.

2 Threat Model

Our threat model is based on the general architecture of the eID Online Authentication outlined in [5] and shown in Figure 1. We differentiate between a *channel* and a *connection*. A *channel* denotes a set of connections which share source and destination entities as well as the cipher suite and, if applicable, credentials, e.g. pre-shared key.

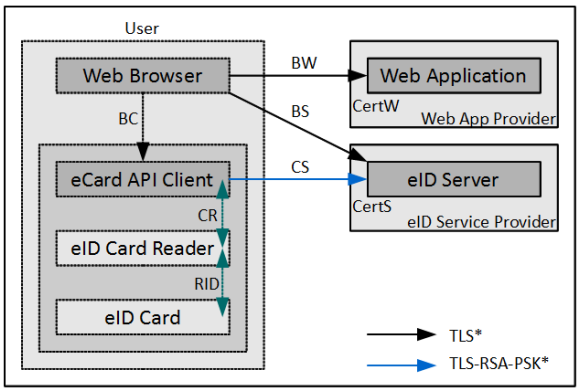

Abbildung 1: eID Service Architecture

The eID Online Authentication provides a secure way to mutually authenticate a web application and an end user. From a security point of view, this has two major advantages. First, the authentication of the web application against the user proves that the user communicates with a benign web site, trying to prevent from service forgery attacks such as phishing. Second, the authentication of the end user enables the web application to *reliably* read personal data from an eID card, such as e.g. name and address, age or a site-specified identifier that could be used as login name.

The most important network channels involved in the eID Online Authentication are depicted in Figure 1 as solid lines. Black lines symbolize channels secured by Transport Layer Security

(TLS [6]), blue lines symbolize channels secured by TLS with RSA-PSK cipher suites [9]. The channel *BW* is the very first channel initiated by the user in order to use a certain web application, e.g. a webshop or a bank. In order to initiate the eID Online Authentication, the user's web browser connects to the eID server using the TLS channel *BS*. The eID server returns a trigger object in *BS* that causes the eCard API client application to connect to the eID server using channel *CS*. The trigger object contains a pre-shared secret as well as a client identity. Listing 1 shows an example of an HTML trigger object for the eID client. Both of these credentials, the PSK client identity and the PSK, are required to successfully establish *CS* using a special set of cipher suites that combine certificate based peer authentication with pre-shared key as key exchange (TLS-RSA-PSK). The trigger object parameter ServerAddress specifies the eID server that will be used for the eID Online Authentication. In addition, the parameter PathSecurity-Protocol in Listing 1 specifies that TLS-RSA-PSK must be used in the channel *CS*.

Listing 1: Example trigger object in the channel *BS*

```
<object type="application/vnd.ecard-client">
 <param name="ServerAddress" value="eid-ref.eid-service.de:443"/>
 <param name="Binding" value="urn:liberty:paos:2006-08"/>
 <!-- RFC 4279 specifies TLS-RSA-PSK -->
 <param name="PathSecurity-Protocol" value="urn:ietf:rfc:4279"/>

 <!-- this is the PSK client identity for the channel CS -->
 <param name="SessionIdentifier" value="54dcf4d212c990c7a768ce51efad"/>

 <!-- this is the PSK for the channel CS -->
 <param name="PathSecurity-Parameters" value="<PSK>c033f52671c..5c36e3759f0cf40837</PSK>"/>
 <param name="SHA256ofSAMLRequest" value="MDEwDQYJYIZIAW...ChnhhAxzs7Cy"/>
 <param name="RefreshAddress"
   value="https://eid-ref.eid-service.de:443/epa/plugin?UEsDBBQACAAIAL2...QAAC8BAA%3D%3D"/>
</object>
```

If all connections are established successfully, Terminal and Chip Authentication [3] are performed resulting in secure messaging between the eID card and the eID server. Thus, on top of the TLS channel *CS*, there is an end-to-end secure messaging channel between the eID card and the eID server. In a successful eID Online Authentication, the personal data is transmitted from the eID card through the secure messaging channel on top of the TLS channel *CS* to the eID server. However, it is usually not (only) the eID server but (also) the web application that needs the personal data (or parts thereof). Thus the personal data is forwarded from the eID server via *BS* and *BW* to the web application. At some points, the Terminal Authentication and the TLS channels are intertwined which will later be described in more detail. This binding aims to prevent intercepting or manipulating the personal data even in the channels *BS* and *BW*.

Our network-oriented threat model is based on the following preconditions:

1. The attacker does not have local system-level access on the user's system.
2. The attacker does neither have local system-level access on the intended eID server nor on the web application server.

3. The attacker can trigger the browser and the client application to connect to a destination specified by the adversary, e.g. via social engineering or DNS spoofing.

4. The attacker is positioned as man in the middle such that packets of any connection between the victim and the web application provider and the eID service provider flow through the attacker's intermediate system.

As our threat model is centered on the network communication, it covers multiple attack goals. As an example, an attacker might want to break any of the channels *BW*, *BS* and *CS* in order to intercept personal data that is transmitted during an online authentication. Another example would be to break the channel *CS* in order to exploit possible weaknesses of the eCard API client application. In addition, although the eID infrastructure is designed to prevent from phishing attacks, an attacker could try to intervene in channel *BW* and perform session hijacking.

We define our threat model to comprise the three channels *BW*, *BS* and *CS* as well as the triggering of the eCard API client *BC* shown in Figure 1. Oepen and Morgner [11] have shown that relay attacks where the attacker is positioned between the eID card API client application and the card reader on the user's system (*CR* and *RID* in Figure 1) can successfully be launched. However, as these attacks require the attacker to have local system access on the user's system, they are out of scope of this work.

In the following, we look into attacks on the network channels based on our threat model.

3 Attack Prerequisites

In order to be able to perform man-in-the-middle attacks on the channels *BW*, *BS* and *CS* between the user's system and the eID server components, several requirements must be met. The channels *BW* and *BS* use TLS with X.509 certificates as required by the eCard API framework [4]. Thus, the attacker must be able to generate these kinds of certificates.

The network channel *CS* between the client application and the eID server is secured by TLS with certain pre-shared key cipher suites, defined in RFC 4279 [9]. In the context of the eID Online Authentication, the channel *CS* must be established using TLS-RSA-PSK cipher suites [4]. The pre-shared key *psk* and the client identity *cid* that are required to establish the channel *CS* are transmitted in advance in the trigger object as part of the previous channel *BS* between the browser and the eID server. Thus, determining the credentials for *CS* either requires brute-forcing them or, in turn, requires the attacker to intercept channel *BS*. This will be covered in more detail in Section 4.2.

At the time of this work, there was no free implementation of TLS-RSA-PSK cipher suites available. Hence, we have developed and provide a patch that implements TLS-RSA-PSK cipher suites in OpenSSL [1]. To our knowledge, this is the first implementation of these cipher suites in a free open source crypto library.

4 Attacks

We break down attacks based on the threat model into two categories.

1. Attacks on the channels *BW* and *BS*
2. Attacks on the channel *CS*

In the following, two kinds of attack phases of man-in-the-middle attacks are distinguished. The first phase is completed, if and only if the man-in-the-middle entity – the TLS proxy – has successfully performed the TLS handshake in both the incoming connection as well as the outgoing connection. At this point, messages of the upper layers can be relayed and modified at will. The second phase is regarded complete, if and only if all messages of the upper layers have successfully and transparently been relayed, effectively resulting in a *successful* eID Online Authentication. This includes the final step where the result of the eID Online Authentication is transmitted from the eID server to the web application server. Furthermore, we define the eID Online Authentication as broken, if and only if the second phase of the man-in-the-middle attack is complete.

We decided to focus on man-in-the-middle attacks for a number of reasons. Cipher degradation attacks where the client's initial ClientHello message is modified by removing strong ciphers do not work against TLS and will result in a handshake error. Given our threat model, an attacker with no system-level access on the user's computer has no way to force certain cipher suites, e.g eNULL, to be used. Most, if not all, current TLS implementations strive to agree on the strongest available cipher suites on both client and server. Only a man-in-the-middle can thus influence the cipher suite agreement. Furthermore, we decided to perform man-in-the-middle attacks because we want to be able to read and modify the communication.

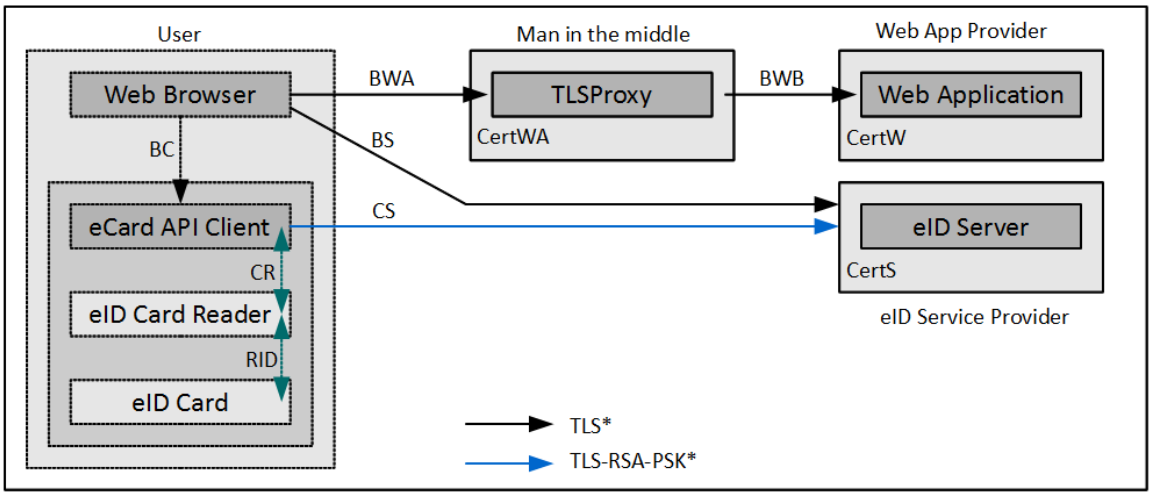

Abbildung 2: Man-in-the-middle attack on *BW*

4.1 Attacks on *BW* and *BS*

According to the eCard API Framework [4], *BW* must be a TLS connection using an X.509-based certificate for server authentication, which is denoted as *CertW* in Figure 1. Intercepting *BW* starts with a man-in-the-middle attack where the attacker tries to impersonate the web application server and proxies all requests to the real web application server, as shown in Figure 2. Therefore, an X.509 certificate, *CertWA*, has to be generated by the attacker which will be used for server authentication of the TLS proxy. Ideally, in order to successfully perform the TLS handshake without user intervention, the browser should successfully verify the X.509 certificate *CertWA*, e.g. *CertWA* might have its certificate chain be rooted in one of the browser's trusted root CAs. Note that we do not aim at discussing the trustworthiness of TLS certificate chains. However, in order to support our approach, let us give two examples. Recent studies have shown that users tend to ignore browser warnings for invalid certificates [12] and additionally the certification processes of SSL may not be conducted with enough rigor [10]. In this scenario, the attacker is successfully positioned between the user's browser and the web application server and can proxy messages between both entities. The first phase of the man-in-the-middle attack is thus completed successfully. Messages of the layers above TLS are relayed through the attacker's TLS proxy and the attacker can modify these messages at will. However, as will be shown later, additional means prevent that personal data is actually transmitted over the intercepted channel. For now, we sum up:

- The channel *BW* is split in two sub-channels, *BWA* and *BWB*, as shown in Figure 2.
- Messages transmitted between the browser and the web application are proxied by the attacker and message contents of the TLS layer, i.e. TLS payload, is available in the attacker's TLS proxy.
- Phase one of the man-in-the-middle attack is completed successfully.

Analogously, a man-in-the-middle attack on *BS* is performed in a similar manner.

4.2 Attacks on *CS*

Intercepting the TLS-RSA-PSK channel *CS* between the eCard API client application and the eID server – see Figure 3 – is similar to the man-in-the-middle attack on *BW* described above. Additionally, the attacker needs the credentials used in the pre-shared key handshake. These credentials, a 256-bit pre-shared key *psk* and a 112-bit client identity *cid*, are generated by the eID server and transmitted to the user's system in channel *BS*. The trigger object *BC* (see Listing 1) provides the eCard API client application with these credentials.

Given our threat model, the attacker does not have local system access on the user's system and is thus required to either brute-force the credentials or break the channel *BS* in order to intercept the credentials for *CS*. Given the fact that an eID server uses high-security modules and provides strong credentials we regard the brute-force approach as infeasible. Hence, given our threat model and its preconditions, in order to intercept channel *CS* the attacker is required to intercept *BS*, too. Once *BS* is broken and the credentials for *CS* are available to the attacker, *CS* can be intercepted.

So far, we have shown how man-in-the-middle attacks can be launched against *BW*, *BS*, and *CS*. In all cases, the TLS handshake is completed successfully, thus the first phase of the man-in-the-middle attack succeeded. At this point, an attacker can relay and modify messages of the upper layers. For example, in case of exploitable software vulnerabilities, an attacker might be able to exploit the eCard API client implementation.

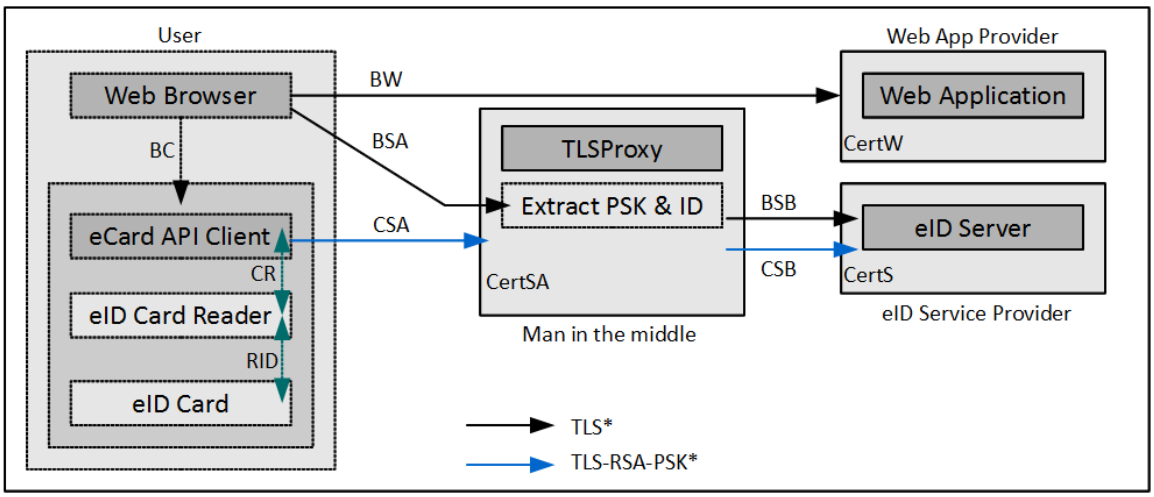

Abbildung 3: Man-in-the-middle attack on *CS*

4.3 Second Phase Failure

However, in theory, none of the above mentioned attacks complete the second phase. This is due to the intertwining of the Terminal Authentication and the TLS channels and it indeed theoretically prevents man-in-the-middle attacks to complete the second phase as described in the following. During an eID Online Authentication, a Terminal Certificate is transmitted as part of the Terminal Authentication [3]. The Terminal Certificate contains the two hash values of the X.509 certificates of the eID server and the web application server, $H_{TC}(CertW)$ and $H_{TC}(CertS)$. As soon as the Terminal Authentication is performed over the channel *CS*, the peer certificates *CertW* and *CertS* are checked against the hash values of the Terminal Certificates, i.e.

$$H(CertW) \stackrel{?}{=} H_{TC}(CertW)$$

and

$$H(CertS) \stackrel{?}{=} H_{TC}(CertS)$$

Due to the fact that the attacker creates its own peer certificate, *CertWA* or *CertSA* or both, with its own public keys, the hash values of the peer certificates of the TLS proxy do not match the ones in the Terminal Certificate, i.e.

$$H(CertWA) \neq H_{TC}(CertW)$$

and/or

$$H(CertSA) \neq H_{TC}(CertS)$$

The eID Online Authentication aborts immediately, if any of the pairs of hashes do not match. Furthermore, the eID client implementation must make sure, that *during and after* a successful online authentication, any subsequent connection in the channel *BW* undergoes this check.

This requirement is currently not stated in the specifications. In practice, the fact that only the very first connection in the channel *BW* undergoes this check (as of February 2011), enables us to man-in-the-middle any subsequent connection in the channel *BW*. We even achieved to break the connection of the channel *BW* where the response of the eID Online Authentication is sent to the web server.

Furthermore, the attacker could modify the hash values of the certificates in the Terminal Certificate:

$$H_{TC}(CertW) := H(CertWA)$$

and/or

$$H_{TC}(CertS) := H(CertSA)$$

However, modifying the Terminal Certificate results in an immediate abort, as the validity of the Terminal Certificate signature is verified by the eID card with the public key of the signer stored in the eID card in secure memory. As a consequence, as soon as the Terminal Certificate is modified, the signature of the Terminal Certificate becomes invalid. As only a very limited number of authorities are trusted root signers of Terminal Certificates (CVCA), it is practically impossible to forge a TC signature. Hence, to sum up, in theory, it is not possible to perform a second phase man-in-the-middle attack. In addition, practical implementations must make sure that these integrity means are applied to all connections of the channel *BW*. Lack thereof will result in successful man-in-the-middle attacks on connections in *BW*.

5 Practical Implications and Recommendations

Interestingly, the set of cipher suites that are allowed to be used for *BS* and *CS* is not restricted, thus the default set of cipher suites of TLS [6] applies. This default set of TLS cipher suites comprises eNULL – a cipher suite with no encryption. As of February 2011, we found that the list of allowed and usable ciphers do contain weak cipher suites such as DES64, IDEA and RC2 or MD5. We suggest to restrict the set of allowed cipher suites for TLS channels in the context of the eID Online Authentication, e.g. to comply with [2], effectively preventing from non-encrypting cipher suites.

Personal data – after being transmitted as part of the secure messaging of the eID authentication – is often displayed on a web page to the user. As an example, if a customer registers with an online shop to create a new account, some if not all of the personal data (name, address, etc.) are displayed on a web page for the user to confirm. While the personal data might have been read and transmitted from the eID card to the eID server using the Online Authentication, on the way back in order to be displayed to the user, this data is transmitted in the TLS channel *BW* with no additional encryption. In combination with a non-encrypting cipher suite or when successfully intercepting connections in the channel *BW*, this enables an attacker to read the transmitted data when it is reflected to the user on the web page. As a guideline, personal data acquired through eID Online Authentication should not be reflected to the user. Fortunately,

some online shops abbreviate the personal data reflected to the user, limiting exposure. However, a successful man in the middle can furthermore read session cookies, effectively allowing session hijacking.

The eID client implementation exposes a certain attack surface even inside the TLS channel *CS*. Whereas in traditional TLS, where connection establishment is handled during the handshake, here the connection is established, verified and possibly torn down only after a couple of messages have been transferred. To be more precise, Terminal Authentication is performed over the established connection in the channel *CS*. At this point, a man-in-the-middle can attack the eID client implementation, e.g. by modifying the TLV-encoded Certificate object. With certain versions of the AusweisApp, removing the Digital Signature element from the TLV-encoded Certificate has lead to an application crash. This underlines the risk that once the TLS handshake is performed, the attacker can attack the applications on the upper layers, client- and server-side, though server-side attacks have not been analyzed here.

6 Related Work

As the eID Online Authentication is actually in use since November 2010, so far, only few people have analyzed actual implementations. Probably the closest work to ours is that of Morgner and Oepen [11]. The authors outline and perform relay attacks which relay access to the eID card remotely over the network. Thus, the location of the eID card and the location of the eID card reader can be separated. However, relay attacks require system-level access on the user's computer. In contrast, our work does not require system-level access, instead it focusses on network-level attacks.

On a theoretical level, Dietrich has analyzed a variety of attacks in [7]. However, none of these attacks target the SAML integration of Extended Access Control. Instead, Dietrich has dealt with man-in-the-middle attacks against Terminal and Chip Authentication.

Eichholz et al. [8] provide an overview of how Extended Access Control can be mapped to Security Assertion Markup Language. Thus, their work is the basis for our work. Additionally, we outline a specific threat model and show at what stage of the communication in the eID Online Authentication, man-in-the-middle attacks fail.

7 Conclusion

In this work, we presented a specific threat model focussed on the network communication in the context of the eID Online Authentication. Our work outlines man-in-the-middle attacks against the three TLS channels that are subject to the eID Online Authentication. The attacks are divided into two phases based on reasonable layers. Furthermore, we show step by step, why – in theory – the attacks finally fail due to the intertwining of the TLS peer certificates with the Terminal Certificate.

However, certain risks remain, especially at the end of phase one where an attacker can easily provide malicious input to the eID client as well as the eID server. These implementations

must make sure, that input is correctly validated. In addition, integrity checks of all channels involved should be applied to all connections of the corresponding channels. The threat model presented in this work can be used to methodically check the communication channels of the eID Online Authentication. It can furthermore be used to develop extended guidelines for implementations, client- and server-sided, in the context of the eID Online Authentication.

Acknowledgements

We thankfully acknowledge support and interesting discussions with Felix C. Freiling, Marian Margraf and Jens Bender. We thank the anonymous reviewers for valuable comments.

Literaturverzeichnis

[1] TLS-RSA-PSK Patch for OpenSSL 1.0.0c, OpenSSL license. http://www.internet-sicherheit.de/service/tools/patches/.

[2] Technische Richtlinie Kryptoverfahren. BSI Technical Guideline BSI TR-02102, June 2008.

[3] Advanced Security Mechanisms for Machine Readable Travel Documents - Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE) and Restricted Identification (RI). BSI Technical Guideline BSI TR-03110, Sept. 2010.

[4] eCard-API-Framework - Protocols. BSI Technical Guideline BSI TR-03112-7, Sept. 2010.

[5] Technische Richtlinie eID-Server. BSI Technical Guideline BSI TR-03130, Oct. 2010.

[6] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878.

[7] C. J. Dietrich. Web-Authentisierung mit dem elektronischen Personalausweis (ePA/nPA). Master's thesis, 2010.

[8] J. Eichholz, D. Hühnlein, and J. Schwenk. SAMLizing the European Citizen Card. In A. Brömme, C. Busch, and D. Hühnlein, editors, *BIOSIG*, volume 155 of *LNI*, pages 105–116. GI, 2009.

[9] P. Eronen and H. Tschofenig. Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). RFC 4279 (Proposed Standard), Dec. 2005.

[10] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The SSL Landscape – A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements. In *Proceedings of the 11th ACM SIGCOMM conference on Internet measurement conference - IMC '11*. ACM Press, 2011.

[11] F. Morgner and D. Oepen. "Die gesamte Technik ist sicher - Besitz und Wissen: Relay-Angriffe auf den neuen Personalausweis, 2010.

[12] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *USENIX Security Symposium*, pages 399–416, 2009.