

# Security analysis of OpenID, followed by a reference implementation of an nPA-based OpenID provider

Sebastian Feld · Norbert Pohlmann

Institute for Internet-Security  
Gelsenkirchen University of Applied Sciences  
{feld | pohlmann}@internet-sicherheit.de

## Abstract

OpenID is an open, decentralized and URL-based standard for Single Sign-On (SSO) on the Internet. In addition, the new electronic identity card (“Neuer Personalausweis”, nPA) will be introduced in Germany in November 2010. This work shows the problems associated with OpenID and addresses possible solutions. There is also a discussion on how to improve the OpenID protocol by the combination of the nPA respectively the Restricted Identification (RI) with an OpenID identity. The concept of an OpenID provider with nPA support will be presented together with its precondition. The added value created by the combination of the two technologies nPA and OpenID in different directions is discussed.

## 1 OpenID as a standard for SSO on the Internet

### 1.1 Problem

Today, users of IT systems in both the private and the business environment have to memorize more and more access information.

In the private environment this arises from the fact that more and more services move into the Internet. The served applications range from e-mail clients and office suites to social networks. There is a login (the claim and the subsequent proof of identity) in almost every service before it can be used. This becomes a problem if a user chooses too short or simple passwords, uses the same password for different services (for convenience) or writes down the passwords.

But even in business environment, employees have to take care of the subject Identity Management (IdM) and its implications. Through the personal use of services an employee will perform various logins often several times a day. Examples are the login to the operating system, to customer databases and e-mail accounts or the use of the corporation’s Internet. A company may establish password policies that define a minimum length for certain passwords or the need to change them at regular intervals. According to experience an increase in security often leads to a decline in user friendliness or efficiency as well. In addition, there are costs resulting from non-

productive time (an employee returns from vacations and forgot the password), or the operation of a user help-desk (a central place to restore forgotten passwords amongst others).

There are different remedy approaches for the problem described. This work deals in particular with the idea of Web Single Sign-On (Web SSO) and the so-called strong authentication. On Web SSO there is only one identifier and a unique authentication using, for example, a strong password. The disadvantage is the single point of failure (the identity manager's service) and the urgent risk of phishing. OpenID is an example of a Web SSO protocol. On strong authentication (also multi-factor authentication), multiple factors like knowledge, possession and property are used to determine identity. A classic example is the use of smart cards with digital certificates. A concrete implementation of this strategy is the eID feature of the new electronic identity card in Germany.

## 1.2 Overview of OpenID

OpenID is an open, decentralized and URL-based standard for SSO on the Internet [ReRe06]. In version 2.0 of the specification (since 2007), a user can freely choose both the identity and the identity manager [ReRe07]. The identification of a user takes place via the proof of the possession of a URL, called OpenID identity.

The great benefits of Web SSO in general, and OpenID in particular is the one-time login at the identity manager (OpenID provider, OP) and the subsequent use of any OpenID-supporting services (relying party, RP). The credentials of a user (client, C) are not longer deposited at many points on the Internet, but only at a central and trusted authority, the OP. Consequently, the digital identity of a user is no longer distributed and redundant, there is only one identifier – the OpenID identity (Identifier, I).

The biggest danger in context of OpenID is the high vulnerability to phishing when using passwords. If an attacker acquires the password of an OpenID identity, all connected services are available to him or her. This can be done, for example, through phishing or by the fact that a user chooses a weak password. Another problem is the possibility of profiling on the part of the OP. The OP knows both the services utilized by the user and the frequency of use and thus could sell these information as user profiles.

## 1.3 Course of the protocol

The execution of OpenID consists of seven steps which are described more detailed below (see Figure 1):

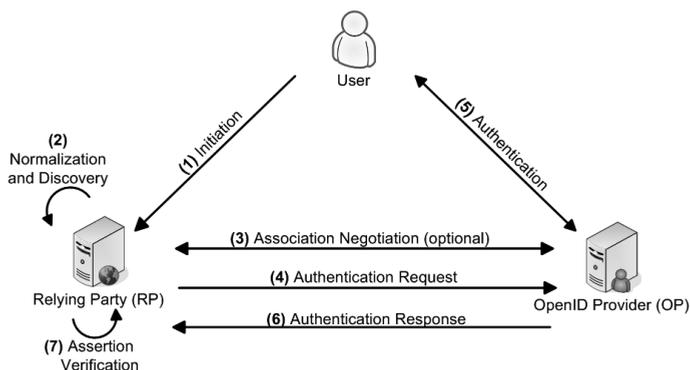


Figure 1: Course of the OpenID protocol

**Initiation** is the transfer of the user-chosen identifier to the relying party which starts the login process. A user calls the website of the service provider (RP), and names only an OpenID identity (the identifier) instead of a user name and password. An example could be *https://openid.internet-sicherheit.de/johnDoe*. The submission of the HTML form to the RP ends the first step.

**Normalization/Discovery** describe the process by which the relying party converts the OpenID identity entered by the user in a standardized form on the one hand and obtains information about the responsible OpenID provider on the other hand. The RP starts with normalizing the identifier entered by the user (see [BeFM05], chapter 6). An example is the addition of a missing schema such as *https://* to *openid.internet-sicherheit.de/johnDoe*. Subsequently, the RP executes the Discovery in which the information needed for generating an authentication request are determined. The XRDS or HTML document identified by the Discovery contains information about the location of the OpenID service on the part of the OP (OP Endpoint URL), the version of the supported OpenID protocol (Protocol Version), the name of the claimed identity (Claimed Identifier) and an alternative representation of the identifier (OP-Local Identifier).

**Association Negotiation (optional)** establishes a communication link with secured integrity between relying party and OpenID provider. RP and OP negotiate a shared secret in order to digitally sign and verify subsequent OpenID messages. If a RP is not capable of creating or saving associations (optionality of this step), the so-called “stateless mode” is used. For this, the OP creates a private secret for signing OpenID messages. The RP verifies messages received through direct communication with the OP (see [ReRe07], chapter 11.4.2).

**Authentication Request** is the request of the relying party to the OpenID provider to authenticate the user. The RP forwards the user’s web browser together with the OpenID authentication request to the OP.

**Authentication** is the actual verification of the user’s identity. The OP checks whether the user is in possession of the OpenID identity and whether he or she wishes to perform the current authentication. The characteristic of the user’s authentication is not specified in the standard ([ReRe07], chapter 3). The responsibility is entirely with the OP, on whose statement a RP has confidence in. The execution of authentication is effectively outsourced. These days the combination of user name and password is a common mechanism for authentication.

**Authentication Response** is the response of the OpenID provider to the relying party together with the statement whether the user's authentication was successful or not. For this purpose, the OP forwards the user's web browser with a positive or negative response to the RP.

**Assertion Verification** is testing the integrity of the received authentication response by the relying party. The RP reviews the OP's response using the previously negotiated association or via direct request (stateless mode). In case of a correct and positive response from the OP the user is successfully logged in to the service of the RP.

## 1.4 Possible fields of application

The protocol OpenID, originally arisen from the "blogosphere", can be used for private or commercial websites and in business environment as well.

Personal websites created with content management systems or blog software such as Drupal, Joomla, TYPO3, WordPress, and the like, can be made OpenID-enabled using plug-ins. From now on, a user can not only use authentication methods offered by the software, but in addition the methods of the applied OpenID provider. The administration of personal websites or blogs can therefore easily be secured with multi-factor authentication if this is offered by the OP.

Commercial service provider on the Internet can offer various authentication methods as well through the integration of an OpenID interface. In addition, the user's inhibition to login is reduced and the registration is accelerated respectively made unnecessary. The service provider offers the user a direct login, in case he or she has an OpenID identity. The chosen OpenID identity is integrated in the list of registered users and also defined as the visible user name. Furthermore, the service provider can request additional user information from the OpenID provider such as the e-mail address. The user can grant or refuse this request.

But even in business environment, the use of OpenID is conceivable. A company defines a closed domain (for example the websites and services of the intranet) and a dedicated identity manager (the OpenID provider). From now on, employees no longer login to each service and each application individually, but centralized using the company-wide OpenID identity.

## 2 Security evaluation of using OpenID

### 2.1 The main threats: Phishing and profiling

The OpenID protocol is highly vulnerable to phishing. In step "Authentication Request" the user is forwarded from the relying party to the OpenID provider via a simple HTTP redirect. There are two basic categories of phishing attacks.

On the one hand phishing is possible by a relying party. A malicious RP does not forward the user to the "correct" OP, but to an imitation which is also under the control of the attacker. The appearance of the original OP can be copied using proxying. The user enters his or her credentials on the fake OP (the phishing happens), whereby a take-over of the OpenID identity through the attacker is possible.

On the other hand a phishing attack is possible by a malicious or compromised URL-host. The user's OpenID identity does not necessarily need to be at the responsible OpenID provider, but can be located at any other host (the URL-host). It is only important that the user controls the URL. An attacker has to exchange the declaration of the responsible OP in the HTML tags of the OpenID identity. In this scenario, the RP sends a user unwittingly to a wrong OP, which can potentially perform a phishing attack.

The use of strong authentication is a remedy for the main threat of OpenID, "phishing". An attacker can no longer intercept the secret between user and OpenID provider, if the user utilizes, for example, the new German identity card.

The creation of user profiles is a further problem in using OpenID. The OpenID provider is a central authority for the user's logins. An OP can potentially monitor the user's activities on the Internet, as it has the knowledge of the services utilized and the frequency of use.

The knowledge of the used services is unavoidable, as the RP requests the OP to perform an authentication. This temporary data is always available. The human factor "trust" is of importance here. A trustworthy OpenID provider should convince users that its central position is not utilized for misuse of information and profiling. An OP should assure that no profiling is made using mandatory policies, terms and conditions and the like. These aspects can be verified via certification after Common Criteria or a publication of the source code, for example.

## 2.2 Additional risks and concerns

Basically, a user has to trust the chosen OpenID provider. Aspects of trust relate to the use of provided personal data, for example. A user should choose an OpenID provider which points out explicitly that the stored data is not used elsewhere. In addition, a user should determine which business model the OP follows. Another aspect of trust is the measure of the services' vulnerability. A user should consider whether the OP's offered security features are in accord with the own security awareness or not. The choice of the OP is an important step because from now on it is responsible for the security of the own digital identity.

Entering personal information is a security-relevant step as well. An OpenID provider can hold various information such as full name, birthday and the like and transmit them to relying parties on user's request. The protocol extension "OpenID Attribute Exchange" [HaBH08] can accelerate the registration process, for example. A user has to decide how much data he or she will provide. One possibility is an indication of numerous data, resulting in a complete user profile for a single digital identity. Alternatively, a user can specify only a pseudonym, if just the functionality of authentication is required.

A relying party may, depending on the implementation of the OpenID interface, be victim of a denial-of-service attack (DoS attack). An attacker declares a large file or a malicious script instead of a regular OpenID identity. The attacker hopes that the RP overstresses and limits or completely refuses the intended service by loading the entire file or running the script. As a measure against the abuse, a relying party can define time and data limits as well as restrict the allowed protocols and ports for the OpenID identity.

To establish a communication link with secured integrity, RP and OP negotiate a shared secret (the association). To avert a man-in-the-middle attack (MITM attack) on OpenID messages sent,

the exchange of the shared secret and, whenever possible, all communication should base on transport encryption (e.g. SSL/TLS).

At each authentication request a RP communicates a kind of URL (the so-called realm) to the OP, for which the request is valid. With this realm a user can determine that he trusts this RP and grants future authentication requests automatically. A malicious RP can effect by specifying an overly general realm (e.g. *http://\*.de*), that henceforth all authentication requests from relying parties with a certain domain (in this case a German top-level domain) are granted automatically. An OpenID provider should restrict the use of the wildcard \* well directed. In addition, it is advisable that the automated trust should not be utilized by users and not be offered by OPs.

The step of the actual authentication is, as described, particularly vulnerable to phishing attacks and profiling.

A positive authentication response can be the target of a replay attack. An attacker intercepts the message of a successful authentication (the redirect) by sniffing. Inserting the message anew causes the authentication of the attacker as the victim. The OpenID specification recommends the use of nonces (number used once) and timestamps as a countermeasure against replay attacks [ReRe07]. An OP integrates nonces into the authentication response in order to make it unique. A RP accepts responses only if the nonce contained is unknown so far. If a RP receives a response with a nonce already used, the message will be discarded suspected of a replay attack. Timestamps can also be used to restrict the period between authentication request and response. Through this, too old answers will be discarded. Furthermore, the period for holding nonces already used is shortened, saving the RP's resources. The fact that an attacker could be "faster" than the victim is a problem. During a MITM attack, an attacker can intercept the victim's redirect to the OP, reject it and execute it instead of the victim (see [TsTs07]).

The step "assertion verification" is of great importance for a relying party, as it checks the integrity of the authentication response. A RP should verify that it accepts no two authentication responses with the same nonce (replay attack). Furthermore, it should determine whether the responding OpenID provider is authorized to provide assertions for the confirmed identity. This leads the RP to perform a new discovery on the identifier of the response. If the OP of the authentication request is equal to the OP of the determined information using discovery, the authentication request is legitimate.

Another attacking scenario focuses on the Domain Name System (DNS). The name resolution is used several times on discovery and the redirections (authentication request and response). If an attacker is capable of, for example, manipulating the cache of the victim, a seemingly correct redirect can lead the user to a copy of the OpenID provider. A phishing attack takes place. The use of strong authentication such as the eID feature of the German new electronic identity card prevents the interception of credentials.

When implementing an OpenID provider the procedure of "recycling" identities of an OP has to be discussed in order to deal correctly with any possible overlap. The services of a user who deletes an OpenID identity at an OP must not be used by another user, who then registers this exact identifier.

## 3 The new identity card (nPA) in Germany

### 3.1 Overview of the nPA

The new identity card (“Neuer Personalausweis”, nPA) will be introduced in Germany on November 1, 2010. It supports the Federal Government’s eCard strategy, which was decided on March 9, 2005 by the Federal Cabinet. Thus, 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) [BSI10b]. 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 [Marg09].

The new ID card has the size of a credit card (form factor ID-1), so it visually and physically differs to the current identity card. In addition, a contactless chip (RF chip) of the interface ISO 14443 is integrated, which communicates by radio with an RF reader. The chip’s three electronic functions are described briefly below.

The first functionality “ePass” consists of the well-known identity determination as with the current ID card. A person attests to another person that he or she is actually the one he or she claims to be. The biometric feature is designed as an exclusive sovereign application in which a digital photograph and optionally two fingerprints can be stored on the card. This biometric identity function together with cryptographic mechanisms and optical security features on the card body ensure an increased protection against counterfeiting [Reis09].

The second functionality “online authentication” represents the electronic identity verification (also: eID feature). A mutual authentication between two communication partners is realized over the Internet, resulting in both parties knowing with whom they communicate. The right on informational self-determination was included because only the user decides whether a service provider can access certain data inside the nPA or not [Reis09]. This feature is intended for eBusiness and eGovernment.

The third functionality “qualified electronic signature” (QES) in accordance with the German Signature Law (SigG) is also intended for eBusiness and eGovernment. A QES represents the equivalent of a handwritten signature in electronic legal and business processes [Marg09]. This functionality must be enabled with costs on the nPA.

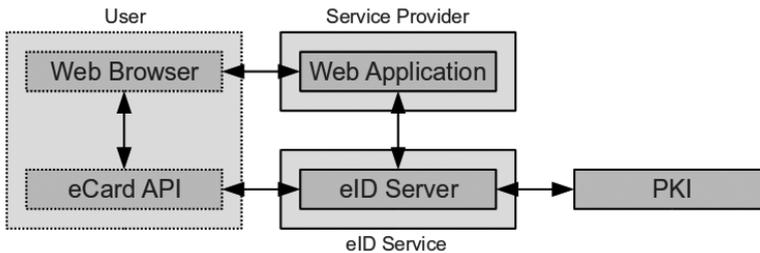
The data’s authenticity and integrity must be ensured both in stored form and during the transmission. The mechanism Extended Access Control (EAC) is to implement these requirements. The protocol aims at ensuring that only authorized readers can access the personal and biometric data of the nPA. The mechanism EAC described by the German BSI is similar to the encryption protocol TLS, and consists of three protocols, “Password Authenticated Connection Establishment” (PACE), “Terminal Authentication” (TA) and “Chip Authentication” (CA). PACE is a procedure that permits a reader to access the radio channel to the nPA after entering a correct PIN. Technically this is an encrypted Diffie-Hellman key agreement (see [BSI10a], chapter 4.2.1). In TA the permissions of the reader are checked by means of a challenge-response procedure (see [BSI10a], chapter 4.4.1). Finally, CA is the process of examining the chip’s authenticity and the implicit authentication of data supplied by the chip. From cryptographic point of view, this is a Diffie-Hellman key agreement with a static chip key (see [BSI10a], chapter 4.3.1). Just if all three

procedures (PACE, TA and CA) of the mechanism EAC are completed, a secure and encrypted end-to-end-channel is established.

When verifying an identity in the electronic world, the check of an document's authenticity is performed using appropriate cryptographic verification methods (see the method EAC just described). This represents the analogy to the visual inspection of the security features in the real world. Instead of comparing the facial image, a secret PIN is required to be entered. The PIN is used for local activation of the ID card's electronic features and therefore is not transmitted over the Internet. The mutual authentication of communication partners is fully completed, after the service provider proved its identity through the so-called terminal certificate. This certificate contains information regarding the validity, the owner, the corresponding public key as well as information about the data the service provider is allowed to read from the nPA's chip. The certificate is issued by a governmental authority ("Vergabestelle für Berechtigungszertifikate", VfB), after a necessity test has confirmed the legitimate interest of personal data.

### 3.2 Course of an online authentication

Below the course of the functionality "online authentication" is explained (see Figure 2), since it will be found in connection with the yet to be explained OpenID provider. It is considered an exemplary scenario in which the user performs a registration to a service provider, including the first and last name is read from the new identity card.



**Figure 2:** Course of an online authentication, based on [BSI10b], Figure 4

The user directs his or her web browser to the service provider's web application and calls a script which offers a registration using the nPA. The service provider is in possession of a valid terminal certificate which states that the first and last name of a user can be read for registration purposes. The need for the information to be read has been demonstrated when applying for the certificate in the appropriate governmental agency. Then the web application contacts the eID server. The eID server is a simple interface for web applications and encapsulates the complexity of electronic authentication [BSI10b]. Specifically, this means that the eID server will perform the concrete communication with the nPA. The service provider instructs the eID server to read the nPA's information using its terminal certificate and transmit them as a response. For this, the eID server sends certain information to the web application that will be forwarded directly to the user's web browser. This information leads to the invocation of the so-called "AusweisApp" (formerly "Bürgerclinet") on the part of the user's PC. AusweisApp is a middleware that implements the communication between card reader, nPA and eID server. This is realized through the eCard-API. The user lays down the identity card on the connected card reader and confirms the current transaction (in this case the reading of first and last name) by entering the secret PIN.

AusweisApp now performs the interaction between nPA and eID server. The information – in this example the user’s first and last name – are subsequently available on the eID server. In the meantime, the service provider’s web application asks at regular intervals, whether there is a response on the request at the eID server or not. If there is a response, the eID server securely transmits the information to the web application. The web application may use the information for the business logic which is, in this example, the continuation of registration using the identity card’s original data.

### 3.3 Recognition via Restricted Identification

The nPA’s specifications provide the recognition of an already registered user. The non-ambiguous identification using the ID card’s serial number is legally not permitted, so the sector-specific identification (Restricted Identification, RI) was introduced.

The RI has two special properties (see [BSI10a], chapter 2.1.5): On the one hand, the RI of a chip is unique within a sector. This means that a user will be recognized without knowing the actual identity. On the other hand, practically it is impossible to connect the RI of a chip between two sectors. This means that accumulated RIs cannot be compared with those of other services and therefore no associations of persons can be made beyond application boundaries.

The protocols CA and TA must have been successfully carried out in order to read the RI. Thus, there is a mutual authentication. The actual protocol for calculating the RI is a key exchange based on Diffie-Hellman algorithm.

## 4 An nPA-based OpenID provider (OP)

### 4.1 Fundamental Concept

An OpenID provider supporting the German new ID card was designed and implemented in the course of this work. The realization is based on two basic ideas.

The first idea covers the proof of the possession of a URL, thus the process of authentication. A user no longer logs in using a user name and password combination prone to brute-forcing and phishing in particular, but by means of strong authentication. Specifically, the nPA’s eID feature is accessed. When registering an OpenID identity, the chip’s Restricted Identification (RI) is linked with the OpenID identity. The only information of the identity card read out – the RI – is used only to recognize the user and never leaves the OP.

The second idea describes the OP’s proxy functionality for the new Identity card. Using the OpenID interface will allow service provider without a terminal certificate to use the eID feature. A user can utilize a strong authentication also at service provider that do not have the necessary financial or organizational resources for the deployment of an eID interface. There are several possible scenarios: “Small” web services without the required resources, closed systems in intranets, but also private applications such as blogs can be designed nPA-compatible using an appropriate OpenID interface.

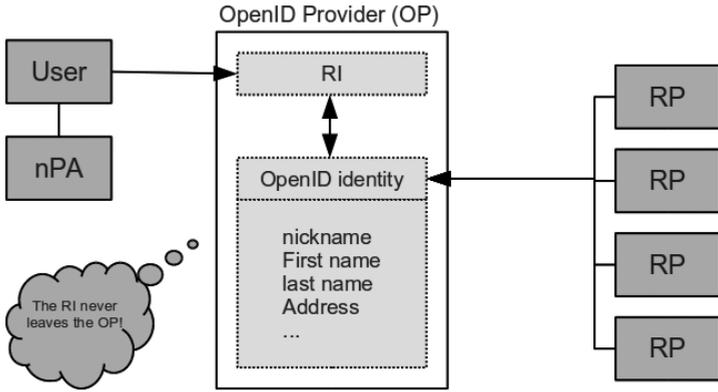


Figure 3: Interface online authentication and OpenID

The schematic connection of the RI to the OpenID identity or rather the interaction of user, OpenID provider and service provider is shown in Figure 3.

### 4.2 OP's communication sequence

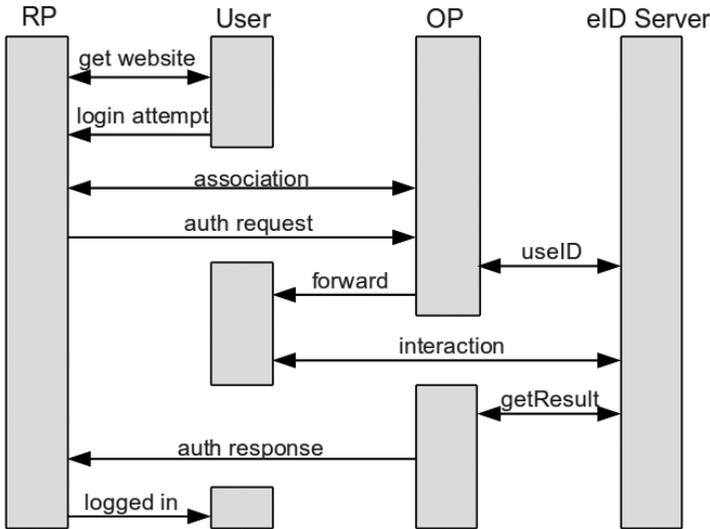


Figure 4: Communication sequence of a login attempt, based on [BSI10b], Figure 6

Figure 4 shows the communication sequence when a user attempts to login to a web service using his or her OpenID identity. The user calls the login form of the service and just enters his or her OpenID identity. After service provider (RP) and OpenID provider (OP) have negotiated a shared secret (the association), the RP sends an authentication request to the OP. The OP contacts the eID server (function "useID") and applies for reading the Restricted Identification (RI) from the user's nPA. The eID server responds with information that the OP will forward directly to the user. The interaction between eID server and user takes places. The user confirms the reading of the RI and enters a secret PIN. The OP checks periodically, whether the results of the requested

action are already available (function “getResult”). If the result is available the determined RI is used to authenticate the user and the OP sends a positive or negative authentication response to the RP. Depending on the result, the service can consider the user to be logged in from now on.

### 4.3 Precondition for user and services

The integration and use of an OpenID provider with nPA support is simple.

From the user’s view only the method of authentication changes. The user has still to prove that he or she actually is in possession of the OpenID identity. For this purpose, a strong authentication based on the nPA is performed instead of requesting a password. The RI read out from the nPA never leaves the OP. The principles of data avoidance and data economy are met.

Service provider need to integrate an appropriate interface into their application in order to use OpenID. Depending on the application’s general implementation (modularity, and the like) the integration is relatively little extensive. There are libraries for various programming languages, so that the OpenID interface can be implemented quickly. There are no additional costs coming up to the service provider, since no other conditions have to be fulfilled beside the integrated interface.

In order to use an OpenID provider with nPA support, a service provider needs not to make other modifications. Such an OP does not only allow “outsourcing” of authentication, but also the use of the nPA’s eID feature via the proxy functionality.

### 4.4 Added value in different directions

OpenID has the overall value that a user has to prove his identity at only a single point. The authority of authentication can be selected more consciously and the effort for a secure configuration can be concentrated. There is only one identifier and one set of credentials to be secured.

The integration of the eID feature as an OP’s authentication method create added value in different directions.

From the perspective of the protocol OpenID the authentication, which is not treated in the specification, is made more secure. The user is no longer able to choose weak passwords (too short, easy to guess, etc.) through the use of the nPA. Furthermore, the biggest problem of OpenID when using a user name and password combination, “phishing”, no longer exists. On the one hand no secret is sent over the Internet when using multi-factor authentication with the nPA. On the other hand there is an authentication of the OpenID provider through the eID feature. Only an OP which is in possession of a terminal certificate is able to read information from the nPA – in this case the Restricted Identification (RI).

A user gets the added value that him or her is not only provided an infrastructure for Web SSO, but also for multi-factor authentication. Since the existing nPA is used, no additional smart cards or readers are required. With the OpenID provider’s terminal certificate, a user can verify the identity of the OP.

One advantage from the perspective of the new identity card is a certain degree of “internationalization” of the eID feature. Basically, the nPA’s use is intended for applications of German

service provider. Service provider without a terminal certificate are not able to use the nPA for authentication, for example. A service provider just has to implement an OpenID interface to use the proxy functionality of an nPA-based OP. From now on, users with a German nPA can login to certain (international) services using that OpenID provider. The service provider can not read information from the nPA (not even the RI), but indirectly use the eID feature via the OpenID provider.

## 5 Outlook

Since the release of OpenID version 2.0 in 2007, there was a rapid development in terms of web standards. Besides OAuth, Facebook Connect or Google Friend Connect, many other “social” protocols have been established in the Web 2.0. The draft of “OpenID Connect” develops a new generation of OpenID. The protocol based on OAuth 2.0 is to simplify the implementation, make OpenID available for desktop applications and introduce other functionalities. It will be seen how the OpenID community will promote and substantiate this draft.

In relation to the new identity card only guesses can be made as well. The Federal Ministry of the Interior (BMI) has supported the introduction of the nPA through a centrally coordinated and an open application test. From November 1, 2010 only the new ID card will be issued. The reaction of the general population and the nPA’s acceptance together with its functions are difficult to assess.

## 6 Summary

The issue of identity management on the Internet is increasingly important in today’s world. And in conjunction with that, the need for verifying the identity must be made as safe as possible. Open standards for Web Single Sign-On, such as OpenID, offer a simple way.

The weaknesses of the OpenID protocol can be compensated by the use of the eID feature of the German electronic new identity card. By combining the two technologies added value are generated in different ways. On the one hand an OpenID provider with authentication using the nPA removes the greatest danger of OpenID “phishing”. On the other hand the use of the nPA can be expanded, virtually internationalized, by the proxy functionality of such an OpenID provider.

The OpenID provider presented in this work will be the first of its kind – or at least one of the first – offering a combination of OpenID with the German new identity card. As currently almost all OPs provide an authentication via user name and password, an OpenID provider with nPA support will be a welcome and also a safe alternative. Transferring a user to a another OP is possible without problems through the decentralized architecture of OpenID.

## References

- [ReRe06] Recordon, David; Reed, Drummond: OpenID 2.0: a platform for user-centric identity management. In: DIM '06: Proceedings of the second ACM workshop on Digital identity management. ACM, 2006, p. 11-16.
- [ReRe07] Recordon, David; Reed, Drummond: OpenID Authentication 2.0 - Final. [http://openid.net/specs/openid-authentication-2\\_0.html](http://openid.net/specs/openid-authentication-2_0.html), 2007.
- [Marg09] Margraf, Marian: Der elektronische Identitätsnachweis des zukünftigen Personalausweises. SIT-SmartCard Workshop 2009, Darmstadt, 2009.
- [BSI10a] 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.
- [BeFM05] Berners-Lee, T.; Fielding, R.; Masinter, L.: RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. <http://www.ietf.org/rfc/rfc3986.txt>, 2005.
- [Reis09] Reisen, Andreas: Die Architektur des elektronischen Personalausweises. 11. Deutscher IT-Sicherheitskongress des BSI, Bonn-Bad Godesberg, 2009.
- [HaBH07] Hardt, D.; Bufu, J.; Hoyt, J.: OpenID Attribute Exchange 1.0 – Final. [http://openid.net/specs/openid-attribute-exchange-1\\_0.html](http://openid.net/specs/openid-attribute-exchange-1_0.html), 2007.
- [TsTs07] Tsyklevich, E.; Tsyklevich, V.: Single Sign-On for the Internet: A Security Story. BlackHat USA, 2007.
- [BSI10b] BSI: Technische Richtlinie eID-Server; Version 1.3. Technische Richtlinie TR-03130, 2010.