Towards secure and standard-compliant implementations of the PSD2 Directive

Tobias Wich[1], Daniel Nemmert[1], Detlef Hühnlein[1]

Abstract: The present article provides a compact overview of the most important requirements of the so-called "Payment Services Directive 2" (PSD2) [Di15], together with the related Regulatory Technical Standard on authentication and communication [Eu17] according to Article 98, and outlines how the pivotal "Access-to-Account-Interface" can be securely implemented based on widely acknowledged international standards.

Keywords: Access control, Strong Authentication, Authorisation, Payment Services Directive 2, Identity Management, SAML, OAuth, OpenID Connect, SOAP, REST, ISO 20022

1 Introduction

The Directive (EU) 2015/2366 [Di15] of the European Parliament and of the Council of 25th November 2015 on payment services in the internal market, also known as the "Payment Services Directive 2" (PSD2), obliges "Account Servicing Payment Service Providers" (ASPSP) (e.g. credit institutions) to allow Third Party Payment Service Providers (TPPs) (e.g. a Payment Initiation Service Provider (PISP) or an Account Information Service Provider (AISP)) to access the payment account of a Payment Service User (PSU) under certain circumstances. With this new requirement, originally introduced under the term "Access-to-Account" (XS2A), the relationship between customers, banks and related service providers is redefined throughout Europe and hence PSD2 may introduce disruptive changes in the European payment market. The newly introduced interfaces are not only subject of strategical and political discussions and initiatives[2], but may reveal new significant technical threats. It seems to be necessary to analyse this topic in a very thorough manner and foster the design, development and implementation of secure and standard-compliant solutions by providing neutral expert guidance.

Against this background the present paper will compile the necessary background and related work in Section 2, before it introduces a proposal for a secure and standard-compliant implementation of the Access-to-Account-Interface in Section 3. Section 4

[1] {tobias.wich, daniel.nemmert, detlef.huehnlein}@ecsec.de, ecsec GmbH, Sudetenstraße 16, 96247 Michelau
[2] See https://www.futureofeuropeanfintech.com, http://www.berlin-group.org/ and https://www.caps-services.com/ for example.

finally draws conclusions and provides an outlook towards possible future developments.

2 Background

2.1 Regulatory Framework

The PSD2 Directive [Di15] provides a continuation of the existing regulatory framework for payment services in Europe [3] and differentiates between **Account Servicing Payment Service Provider** (ASPSP) (see Article 4 (17) and (12)), **Payment Initiation Service Provider** (PISP) (see Article 4 (18)and Article 66) and **Account Information Service Provider** (AISP) (see Article 4 (45) and Article 67).

Since PISP and AISP are permitted to access the payment account of the Payment Service User (PSU) established at the ASPSP, they are also referred to collectively as "Third Party Provider" (TPP).

Under **Article 66** ("Rules on access to payment account in the case of **payment initiation services"**), payers have the right to use a PISP to trigger a payment, which must be authorised by the PSU using strong authentication.

According to **Article 67** (Rules on access to and use of payment account information in the case of **account information services**), a PSU has the right to access a payment account through an AISP.

According to **Article 97** (Authentication), a **strong customer authentication** is required if the payer accesses his payment account online, triggers an electronic payment process, or undertakes, via a remote access, an act which involves the risk of fraud in payment transactions or other misuse. In the case of a remote payment operation, the elements "amount" and "payment recipient" must be dynamically incorporated into the data submitted for authentication.

In accordance with Article 97 (3), PSPs must take appropriate security precautions to protect the confidentiality and integrity of the personalized security features. Article 97 (5) stipulates that the ASPSP needs to allow TPPs to rely on authentication mechanisms that the ASPSP itself provides.

It is important to emphasize that the requirement of Article 97 (5) PSD2 implies the classical "triangular relationship" between the PSU (User), the ASPSP (Identity

[3] See [Di07], [Di09], [Re09], [Di11] and [Re15].

Provider) and the TPP (Service Provider) as utilized in federated identity management[4] and depicted in Section 3.1.

More detailed considerations on the design of these interfaces can be found in Section 3, which in addition to the strong authentication in accordance with Article 97 must also provide the necessary business functionality in accordance with Articles 66 and 67.

In accordance with **Article 98**, the European Banking Authority (EBA), in cooperation with the European Central Bank (ECB), shall develop **Regulatory Technical Standards** (RTS) (cf. [Eu17]), which in particular specify the requirements of the strong customer authentication process under Article 97. Additional standards are required for any exceptions to the requirement of strong authentication, the protection of the confidentiality and integrity of the personalized security features, and "the requirements for common and secure open standards of communication for the purpose of identification, authentication, notification, and information, as well as for the implementation of security measures, between account servicing payment service providers, payment initiation service providers, account information service providers, payers, payees and other payment service providers." [Di15]

2.2 ISO 20022 (Universal financial industry message scheme)

Because of the requirement that the "dedicated interface uses ISO 20022 elements, components or approved message definitions, for financial messaging" in Article 28 (3) of the EBA Draft [Eu17], it seems to be worthwhile to recall the most important aspects of this standard with respect to the Access-to-Account-Interface. ISO 20022 [ISO13] is an international standard developed by the ISO Technical Committee TC68 Financial Services, which consists of multiple parts and describes a common platform for the development of messages[5] for the financial service industry. It comprises a syntax-independent meta model, a corresponding UML profile, guidelines for modelling, stipulations for the generation of XML schemata and ASN.1 modules and a rich catalogue of messages for various financial sector business domains, which in particular includes payments[6].

Among the various payment-related messages, the following two messages seem to be particular relevant with respect to PSD2:

- `CreditorPaymentActivationRequestV06` (pain.013.001.06), because it contains structures for the payer (`Dbtr`), the payment account (`DbtrAcct`), the payment receiver (`Cdtr`), the account of the recipient (`CdtrAcct`), the purpose

[4] See [HRZ10] and [Hü14] for example.
[5] See https://www.iso20022.org/catalogue_of_messages.page
[6] See https://www.iso20022.org/payments_messages.page

(`Purp`) and the amount (Amt), as required for Article 66 PSD2 and

- `BankToCustomerAccountReportV06` (camt.052.001.06), because it allows to address accounts (`Acct`), to specify time periods (`FrToDt`) and account report structures (`Rpt`) as required for Article 67 PSD2.

2.3 Identity, Access and Authorisation Management Technologies

Identity and Access Management (IAM) technologies can only be chosen depending on the technologies they ought to protect. In the case of web technologies nowadays the two major API paradigms are SOAP [Gu07] and REST[7]. Technologies such as CORBA[8], .NET Framework Remoting[9] and XML-RPC[10] never gained widespread use in the open web and are therefore not considered further.

When designing a system like PSD2 where accredited TPPs interact with ASPSPs, there is basically no way around federated Identity Access Management (IAM) systems, if manageability of user identities and transactions is an essential goal.

The **Security Assertion Markup Language** (SAML) [Ca05a] is a family of standards, which has been developed by the OASIS Security Services Technical Committee and defines the syntax and semantics for XML-encoded assertions about authentication, attributes and authorisation together with related protocols that convey such assertions and the binding of these protocols to various transfer protocols. The different versions of SAML have been influenced by previous work in IETF RFC 2903 [La00] and projects like the Liberty Alliance and Shibboleth[11]. The current version of SAML is version 2.0 and comprises various parts, which define Assertions and Protocols, Bindings, Profiles, Metadata, Authentication Contexts, Conformance Requirements and last but not least Security and Privacy Considerations. In order to integrate SAML with SOAP, a set of specifications titled WS-Security has been developed by OASIS. It furthermore specifies message integrity via signatures and confidentiality via encryption. Strongly related to WS-Security are the specifications WS-Trust, defining a Security Token Service (STS) from which assertions can be obtained for machine to machine scenarios, and WS-Policy, defining the necessary means for a secure communication between the entities. The protocols resilience against token theft can be raised significantly with the Holder-of-Key profile [KS10], which binds the SAML assertion to a client-authenticated TLS channel, as shown in [MMS14]. Similar approaches which do not require the distribution of X.509 client certificates have been discussed in [AWZ10], [SLG10] and [BHS08],

[7] https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest
[8] http://www.omg.org/spec/CORBA/3.3/
[9] https://msdn.microsoft.com/en-us/library/kwdt6w2k(v=vs.100).aspx
[10] http://www.xmlrpc.com/
[11] http://www.projectliberty.org/ and https://shibboleth.net/

however missing implementations in the TLS client and server components have prevented their practical deployment so far.

The security of SAML 1.0 was analysed in [Gr03] and the discovered flaws led to recommendations in version 2.0 of SAML (cf. [LM05]). Additional vulnerabilities of the Artifact Profile have also been addressed in [Gr03]. A security analysis for a Liberty-enabled client can be found in [PW03].

A general treatment of security aspects related to the current version of SAML is available in [HPM05]. A formal security analysis of the Web Browser SSO Profile in SAML 2.0 appeared in [Ar08] and revealed a flaw in the SAML-implementation of Google Apps. Other steps towards providing security proofs for browser based protocols can be found in [Ga08], [GPS05] and [FKS14]. In [So12] it was shown that many SAML implementations were susceptible against signature-wrapping attacks.

Authorisation statements in SAML 2.0 can be realized with the **eXtensible Access Control Markup Language** (XACML) [Ri13]. XACML implements access control with the powerful Usage Control (cf. [SP03]) paradigm. The authorisation statement is encapsulated in the SAML identity assertion and thus protected from manipulation. This coupling also binds the authorisation statement to the identity of the user granting the permissions to the service provider. In case access control definitions must be exchanged over protocols which are not based on XML, then the JSON Profile of XACML (cf.[Br14], Chapter 4) might be a better choice than the XML based message variants defined in the core specification.

OpenID Connect (OIDC) [Sa14a], the successor of OpenID 2.0, which is developed by the OpenID Foundation is a lightweight alternative to SAML 2.0 focussing especially on the modern web and mobile usage and has been finished in the beginning of 2014[12]. Additionally to the core it comprises specifications for discovery, dynamic registration, session management, and log-out via front- and back-channel. Work is also ongoing for federated trust establishment with signed metadata. OIDC is based on the OAuth 2.0 Authorization Framework [Ha12]. This implies that OIDC is directly compatible to OAuth 2.0, meaning an OIDC based system can deal with authentication as well as authorisation directly. On the other hand, an OAuth 2.0 based system is not necessarily compatible to OIDC, as this would require signed JSON Web Tokens (JWT) [JBS15b] as ID tokens whereas plain OAuth 2.0 uses opaque access tokens. The use of JWTs in OAuth 2.0 is defined for client authentication and authorisation grants in an auxiliary specification [JCM15].

OIDC provides two main modes of operation, called “flows”: the “server-flow” (or Authorization Code Grant) and the “client-flow” (or Implicit Grant). While the first one is used for web-applications which receive the identity and access token by their server

[12] https://openid.net/2014/02/26/the-openid-foundation-launches-the-openid-connect-standard/

side logic, the client-flow is for JavaScript-based applications, which run completely in the browser. The security of OIDC has been analysed with formal methods in [FKS17], a more systematic treatment of OAuth 2.0 can be found in RFC 6819 and attacks against OIDC have been systemized in [Ma17]. The security of an OIDC-based system highly depends on details of the implementation and naive implementations may be attacked by Cross-Site-Scripting (XSS), Cross-Site-Request-Forgery (CSRF) and Man-in-the-Middle (MITM) attacks for example. Furthermore it seems to be worthwhile to point out that OIDC and OAuth 2.0 as of today only uses "bearer tokens", requiring only the possession of the token. The more secure bindings based on the still unfinished TokenBinding [Po17] are in a rather early stage of development [JB17]. TokenBinding defines how an asymmetric key pair is bound to the TLS channel between the user and the server. When the signed ID token contains the public key of the user, the comparison of the key established over the TLS channel against the key in the token establishes a strong binding to the TLS channel and prevents misuse of stolen tokens.

3 Towards a secure and standardized Access-to-Account interface

In this section we will derive use cases from the requirements of the PSD2-Directive and develop a secure technical design for the Access-to-Account-interface based on widely accepted international standards.

3.1 Use Cases

The use cases for an Account-to-Account interface compliant with the PSD2 directive can be derived from Articles 66 and 67 PSD2 and will be grouped according to these two articles.

Article 66 provides **rules for the initiation of a payment** by a payer (i.e. the PSU) to a designated payee. With this information available, three use cases can be derived. Before any payment can be executed with a PISP involved, the *modalities of the payment* have to be managed. A PISP may not hold the funds it received by the payer for any amount of time and has to relay them to the payee as fast as possible. The information contained in the executed payment may only contain the user and the issuer of the security credential of the payer not the credential itself. Information about the payment is only to be transmitted via secure channels (see sections 3.2 to 3.4). Information about the payer, which is known to the PISP, may only be transmitted to the payee with explicit consent by the payer. In line with the principles "data minimisation" (Article 5 (1) (c)) and "purpose limitation" (Article 5 (1) (b)) of the European General Data Protection Regulation (GDPR) [Re16], the PISP may only request information from the payer that is necessary to execute the payment. Additionally, the PISP informs the user about the

precise purpose the requested information is used for. The PISP may not alter the information of the payment in any way. To *initiate a payment* the PISP has to authenticate itself and the PSU at the respective ASPSP, before the PSU is able to give its explicit authorisation of the execution of the payment via a PISP to the designated payee. The ASPSP also has to provide all information required for the execution of the payment immediately after receiving the request from the PISP and treat payment execution orders by a PISP without prejudice. Article 64 (1) PSD2 provides a third use case with a payment transaction being authorised by the payer *after the execution* of the payment has taken place, which is the third possible use case concerning Article 66. The third use case is not explicitly stated in Article 66, but rather derived from Article 64 (1).

Two additional use cases are necessary for the access to a payment account to **retrieve account information** (Article 67). The first use case is the *authorisation of the AISP*. As with the authorisation of payment initiation services in Article 66, the AISP service has to get the explicit consent from a PSU to be able to provide its service. It also has to take care of the personalised security credentials and has to authenticate itself at the ASPSP. An AISP can only request information that is not considered sensitive or privileged user or payment information. In accordance with data protection regulations like the GDPR, the data accessed by the AISP may only be used, accessed or stored to provide the service the user gave his explicit consent to. After a successful authorisation of the AISP, the second use case allows the AISP to *request the relevant account information* at the ASPSP.

3.2 Solution Outline

The present section outlines a solution for implementing the use cases above based on widely acknowledged messaging patterns and standards. From the use cases above, it is already clear what functionality must be provided and also which services access it. From this information an authentication interface, an account information interface, and a payment processing interface, all provided by the ASPSP, can be derived.

The authentication interface is not accessed by the TPP (PISP or AISP) directly, but provides the authentication and authorisation information of the PSU related to the respective payment service. The remaining interfaces are used by the payment services and have varying requirements regarding freshness of the authentication and immediate user participation. This means that some are transaction based (e.g. payment processing) and some may be used repeatedly (e.g. account information retrieval) by the payment service. This fact must be reflected in the authentication information issued by the payment provider.

The interface to strong authentication as set out in Article 97 PSD2 can be decoupled from the other technical interfaces in accordance with Articles 66 and 67 by the use of a

federation protocol such as SAML or OpenID Connect.

The access of the technical interfaces requires an appropriate identification by the TPP. Article 29 (1) [Eu17] requires that for this purpose qualified certificates for electronic seals according to Article 3 (30) of the eIDAS Regulation [Re14] or qualified certificates for website authentication pursuant to Article 3 (39) eIDAS are required. Since the required identification of the TPP could only take place with certificates for website authentication (if the ASPSP would initiate the connection setup or "call back" the requesting TPP) it is clear that one needs to use qualified certificates for electronic seals according to Article 3 (30) eIDAS for the identification of the TPP. An advanced electronic seal according to Article 3 (26) can be created using a digital signature according to [ET16c] in the case of an XML-based message or in accordance with [JBS15a] in the case of a JSON-based request message. It is worth mentioning that the electronic seal required for identification does *not* have to be a qualified electronic seal in accordance with Article 3 (27), but can also be produced *without* a qualified electronic seal creation device according to Article 39 of the eIDAS Regulation. An alternative to using the electronic seal directly in the authentication to the service, it could be used in the metadata exchange (cf. [Ca05b]) and the dynamic registration specifications (cf. [SBJ14] & [Sa14b]) of the individual Access Management solution to produce more lightweight and short lived secrets.

As already mentioned above, Article 97 (5) PSD2 and Article 27 (3) (a) [Eu17] imply the "triangular relationship" between the PSU (User), the TPP (Service Provider), and the ASPSP (Identity Provider) as shown in Figure 1. The process comprises the following steps:

(1) The PSU requests a specific action at the TPP.
(2) The TPP sends a corresponding request to the ASPSP. For a payment initiation according to Art 66 PSD2 this may contain transaction information. Alternatively the transaction information could be transported to the ASPSP in the next step.
(3) The ASPSP redirects the PSU to the authentication endpoint in order to ask for her consent for a specific transaction.
(4) In this step the strong authentication of the PSU and the authorisation of the requested action takes place.
(5) After the successful user consent, authentication and transaction authorisation, a corresponding response is returned to the TPP.
(6) The TPP now may invoke the business interface corresponding to the use case under consideration at the ASPSP according to Article 66 or 67 for example. In case of payment transaction, the response may contain the integrity protected and possibly encrypted transaction data. Non-transaction driven interfaces such as account information retrieval may be used after step (6).

(7) Finally the PSU receives an acknowledgment of the process.

Pursuant to Article 27 (4) [Eu17] the technical interfaces offered by the ASPSP must be based on international or European standards and in accordance with Article 28 (3) [Eu17] – at least in the case of a dedicated interface – must use elements, components or messages from ISO 20022.

Among the currently relevant international standards, that enable strong authentication in a "triangular" relationship as depicted in Figure 1, in particular the aforementioned Security Assertion Markup Language (SAML) Version 2.0 and the OAuth 2.0 based OpenID Connect are noteworthy.

While the two protocols differ in technical details, both support the architecture outlined in Figure 1 and therefore could basically be used as the basis for the realisation of PSD2-specific interfaces. Thus, it would be theoretically conceivable to specify a corresponding PSD2 interface on the basis of SAML as well as on the basis of OAuth 2.0 / OpenID Connect, whereby both SOAP- and REST-based web services can be used for the technical interfaces.

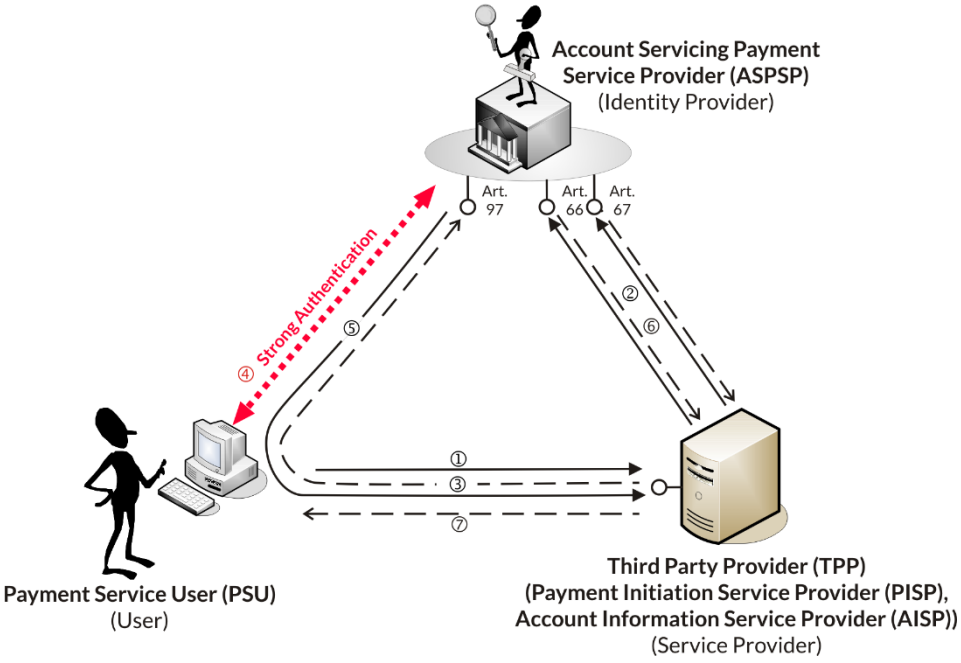

Figure 1: Co-operation between PSU, TPP and ASPSP according to the PSD2-Directive

3.3 SAML-based Architecture

Concerning the decision about the choice of the protocol, there have been already requirements articulated by the German Banking Industry Committee [13] who would appreciate a single uniform standard developed across Europe. Since Article 28 (3) [Eu17] requires the use of elements, components or message types from ISO 20022, a strict interpretation of this requirement would point to a completely XML- and therefore SAML-based solution.

In contrast to the usual three-party SAML scenario, a fourth party which provides the interfaces for the business services (i.e. ASPSP) has to be considered. In general SAML assertions are consumed by the Service Provider, i.e. the TPP, only. In the case of PSD2, the main purpose is to obtain an authorisation from the User in order to execute a business process in one of the technical interfaces. While it is in principle possible to forward a SAML assertion if the destination checks are loosened up, SAML V2.0 Condition for Delegation Restriction Version 1.0 (cf. [LH09], Sec. 2.2.1) provides a stricter way by defining two variants (Forward and Delegation) supporting the delegation needed in PSD2 in a standard compliant way. The specification does not require a specific technical method to relay the assertion which makes it possible to use the SAML assertion as specified in WS-Security when using SOAP. When using a REST based API, an Authorization HTTP header with a custom auth-scheme (cf. [FR14], Sec. 2.1) similar as defined for OAuth 2.0 [JH12] can be used. The SAML delegation specification defines the new type `DelegationRestrictionType` which can be used to name entities which are authorised to delegate the SAML Assertion. While the "Forward mode" is a simple relay of the assertion, the "Delegate mode" uses the Identity Provider (IdP) to obtain a new assertion with the requesting service added as a delegate. While needing more effort on the side of the IdP, the "Delegate mode" has the benefit of being more concise about the addressee of the assertion and thus makes it possible to validate the assertion at each SP (e.g. in step (5) the TPP and in step (6) the ASPSP) without modifying the subject validation mechanism.

When considering the use of SAML Assertions as tokens in an API call, it must be noted that the assertion acts, besides being an authentication token, also as an authorisation token. With that in mind it becomes apparent, that the assertion must also contain the information which interfaces are allowed to be called on behalf of the user. Furthermore, with the transaction based services defined in Section 3.2 it becomes apparent, that it must also reflect the grant of the user for a specific transaction. As discussed in Section

[13] [De16]C1 O1 demands: "There must be a single interface for technical communication that is standardized throughout Europe and is uniform for all third-party services and application scenarios." (translated from German)

2.3, SAML 2.0 has the possibility to express authorisation statements with XACML (cf. [Hu05], Sec. 8.5). The transaction and interface access definitions are then encoded as special SAML Attributes and can be used by the ASPSP to check the legitimacy of the request with an existing XACML system, or a minimal subset of an XACML system supporting only the authorisation statements issued by the authentication interface.

A further distinction between transaction and long term access is the fact that assertions in the latter case may be used for a fixed amount of time while the former typically have a one-time use limit. In SAML replay attacks are solved by obliging the receiver of the assertion to maintain a set of IDs of consumed assertions for the time of their validity (cf. [Hu05], Sec. 4.1.4.5).

Article 66 PSD2 for example states that the payment initiation service provider shall "ensure that the personalised security credentials of the payment service user are not, with the exception of the user and the issuer of the personalised security credentials, accessible to other parties and that they are transmitted by the payment initiation service provider through safe and efficient channels." Article 67 PSD2 provides an analogous section for information service providers.

3.4 OpenID Connect-based Architecture

The SAML-based architecture shown in the previous section used SAML Assertions as an authorisation grant. While this is well within the defined semantics of the SAML 2.0 protocol specifications, this approach needs careful consideration about how the assertion is validated and in which context it is accepted. OAuth 2.0 [Ha12] on the other hand defines its goal as follows: *"The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf."* This problem description matches exactly the goals of the general solution outline from Section 3.2, when the TPP takes the role of the third party application and the ASPSP the role of the HTTP service. In order to also provide secure user authentication directly to the TPP, OpenID Connect (OIDC), which is built upon and is compatible to OAuth 2.0, should be used for the reasons outlined in Section 3.2.

OIDC provides different modes of operations, which are based on the OAuth code and implicit flow for different scenarios. These modes are identified by the `response_mode` and `response_type` parameters. For the same reasons requiring confidentiality with the SAML architecture, the identity attributes must not be transmitted in plain to the TPP. As a mechanism similar to the SAML Artifact Binding is not possible with the modes available in OIDC due to the required ID Token in every code flow based mode (cf. [Me14], Sec. 5), the only way to achieve that goal is to

encrypt the messages. The OIDC core specification allows the messages ID Token, UserInfo Response, and Request Object to be encrypted via JSON Web Encryption (JWE) wrapping the messages. Transaction data sent by the TPP is also covered by the signature and encryption if a Request Object (cf. [Sa14a], Sec. 6.1) is used rather than URL parameters. The transaction data itself could be transmitted as an additional XACML based parameter in the request. In order to better align with the REST style nature of OIDC, the JSON Profile of XACML (cf [Br14], Chapter 4) could be used. Using claims parameters containing the transaction data is not recommended, as unknown claims values must be ignored silently by the OpenID Provider (cf. [Sa14a], Sec. 5.5.1). The fact that OIDC is based on an authorisation protocol is reflected in the usage of two different tokens. First there is the ID Token, which is used to authenticate the PSU against the TPP, and second the Access Token which is used to access restricted resources (i.e. the payment services). The access token (after it is obtained by the TPP) is sent as an HTTP Authorization Header in the payment service request. OAuth 2.0 defines no other methods to transmit the authorisation information, which encourages a REST style API interface rather than a SOAP based interface. To address the single transaction case, the authorisation statement created during the user authentication at the ASPSP can be distributed to the interface endpoints either directly via an internal mechanism, or encoded in the Access Token. In the latter case, the Access Token, which is opaque to the TPP, could be an encrypted JWT similar to the ID Token confidentiality mechanism described above.

4 Conclusion and Outlook

The Payment Service Directive 2 (PSD2) has the potential to introduce disruptive changes to the European payment market. The directive regulates the access to user identities and payment services for Payment Service Providers. By requiring strong user authentication it lays the foundation for changes even beyond the financial industry, if the strong authentication provided by the Account Servicing Payment Service Providers (ASPSP) is opened up to be used by organisations and companies outside of the financial industry on a voluntary basis or as might be required by a future regulation.

The technical solutions outlined in this paper use SAML 2.0 or OpenID Connect, which are well researched and very flexible standards for user authentication and access management to payment services. The two solutions also represent the different approaches of the standards. The SAML 2.0 ecosystem provides a variety of composable specifications. While this is extremely flexible and open, it also requires careful consideration of the features and extensions used in order to maintain the targeted security properties. OpenID Connect on the other side represents a more monolithic framework. It is possible to extend certain points, but there are by far less extensions available and as it has been shown, these are also not needed for the PSD2 system. It is

therefore much easier to define the system on top of OAuth 2.0 / OpenID Connect than it is on top of SAML 2.0.

Further research on the PSD2 implementation architectures should especially comprise secure channel bindings with Holder-of-Key or the currently emerging TokenBinding [JB17]. Another key point of a distributed PSD2 system is the trust relationship between ASPSPs and TPPs, which is envisioned to use qualified certificates for website-authentication or electronic seals. While this work only touches the topic briefly, a more thorough treatment which also covers metadata exchange and dynamic discovery using electronic seals and a common European repository or set of trusted lists has yet to be developed.

References

[Ar08] Armando, A., Carbone, R., Compagna, L., Cuellar, J., Tobarra, L.: Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. Formal Methods in Security Engineering 2008. ACM-Press, 2008.

[AWZ10] Altman, J., Williams, N., Zhu, L.: RFC 5929. Channel Bindings for TLS. Internet Engineering Task Force (IETF), 2009. https://tools.ietf.org/html/rfc5929.

[Br14] Brossad, D.: JSON Profile of XACML 3.0 Version 1.0. OASIS Committee Specification 01. OASIS, 2014. https://docs.oasis-open.org/xacml/xacml-json-http/v1.0/cs01/xacml-json-http-v1.0-cs01.html.

[BHS08] Bruegger, B. P., Hühnlein, D., Schwenk, J.: TLS-federation – a secure and relying-party-friendly approach for federated identity management. In Proceedings of BIOSIG 2008: Biometrics and Electronic Signatures. LNI 137. Darmstadt 2008. Köllen Verlag, Bonn, pp. 93-104, 2008.

[Ca05a] Cantor, S., Kemp, J., Philpott, R., Maler, E.: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS, 2005. https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[Ca05b] Cantor, S., Moreh, J., Philpott, R., Maler, E.: Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard. OASIS, 2005. https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[De16] Anforderungen an eine Datenschnittstelle für Drittdienste. Finale Version. Deutsche Kreditwirtschaft, 2016. https://die-dk.de/media/files/2016-02-22_DK_-_Whitepaper_-_Requirements_data_interface_final_1.2_de.pdf .

[Di07] Directive 2007/64/EC of the European Parliament and of the Council of 13 November 2007 on payment services in the internal market amending Directives 97/7/EC, 2002/65/EC, 2005/60/EC and 2006/48/EC and repealing Directive 97/5/EC. Official Journal of the European Union, 2007. http://data.europa.eu/eli/dir/2007/64/oj.

[Di09] Directive 2009/110/EC of the European Parliament and of the Council of 16 September 2009 on the taking up, pursuit and prudential supervision of the business of electronic money institutions amending Directives 2005/60/EC and 2006/48/EC and repealing Directive 2000/46/EC. Official Journal of the European Union, 2009. http://eur-lex.europa.eu/eli/dir/2009/110/oj.

[Di11] Directive 2011/83/EU of the European Parliament and of the Council of 25 October 2011 on consumer rights, amending Council Directive 93/13/EEC and Directive 1999/44/EC of the European Parliament and of the Council and repealing Council Directive 85/577/EEC and Directive 97/7/EC of the European Parliament and of the Council. Official Journal of the European Union, 2011. http://eur-lex.europa.eu/eli/dir/2011/83/oj.

[Di15] Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC. Official Journal of the European Union, 2015. http://data.europa.eu/eli/dir/2015/2366/oj.

[ET16c] ETSI EN 319 132-1. Electronic Signatures and Infrastructures (ESI); XAdES digital signatures Part 1: Building blocks and XAdES baseline signatures. European Telecommunications Standards Institute, 2016. http://www.etsi.org/deliver/etsi_en/319100_319199/31913201/01.01.01_60/en_31913201v010101p.pdf.

[Eu17] Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2). European Banking Authority, 2017. https://www.eba.europa.eu/documents/10180/1761863/Final+draft+RTS+on+SCA+and+CSC+under+PSD2+%28EBA-RTS-2017-02%29.pdf.

[FKS14] Fett, D., Küsters, R., Schmitz, G.: An Expressive Model for the Web Infrastructure: Definition an Application to the BrowserID SSO System. arXiv, 2014. https://arxiv.org/abs/1403.1866.

[FKS17] Fett, D., Küsters, R., Schmitz, G.: The Web SSO Standard OpenID Connect: In-Depth Formal Security Analysis and Security Guidelines. arXiv, 2017. https://arxiv.org/abs/1704.08539.

[FR14] Fielding, R., Reschke, J.: RFC 7235. Hypertext Transfer Protocol (HTTP/1.1): Authentication. Internet Engineering Task Force (IETF), 2014. https://tools.ietf.org/html/rfc7235.

[Ga08] Gajek, S.: A. universally composable framework for the analysis of browser-based security protocols. In J. Baek, F. Bao, K. Chen, X. Lai (Ed.), Provable Security – Second International Conference, ProvSec 2008. 5324 of Lecture Notes in Computer Science, pp. 289-297. Springer, Shanghai, 2008.

[Gr03] Groß, T.: Security Analysis of the SAML Single Sign-on Browser. Annual Computer Security Applications Conference, December 8-12, 2003. Aladdin Resort & Casino Las Vegas, Nevada, USA, 2003.

[GPS05] Groß, T., Pfitzmann, B., Sadeghi, A.-R: Browser model for security analysis of browser-based protocols. ESORICS: 10th European Symposium on Research in Computer Security. 3679 of Lecture Notes in Computer Science, Springer, Berlin, pp. 489-508, 2005.

[Gu07] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., Frystyk N. H., Karmarkar, A., et al.: SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation. World Wide Web Consortium (W3C), 2007. https://www.w3.org/TR/soap12/..

[Ha12] Hardt, D.: RFC 6749. The OAuth 2.0 Authorization Framework. Internet Engineering Task Force (IETF), 2012. https://tools.ietf.org/html/rfc6749.

[HPM05] Hirsch, F., Philpott, R., Maler, E.: Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard. OASIS, 2005. https://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf.

[Hu05] Hughes, J., Cantor, S., Hirsch, F., Mishra, P., Philpott, R., Maler, E.: Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard. Oasis, 2005. https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[HRZ10] Hühnlein, D., Roßnagel, H., Zibuschka, J.: Diffusion of Federated Identity Management. Sicherheit 2010: Sicherheit, Schutz und Zuverlässigkeit. LNI 170, Köllen Verlag, Bonn, pp. 25-36, 2010. http://www.ecsec.de/pub/Sicherheit2010.pdf.

[Hü14] Hühnlein, D., Wich, T., Schmölz, J., Haase, H.-M.: The evolution of identity management using the example of web-based applications. Information Technology 56(3), pp. 134-140, 2014. http://ecsec.de/pub/2014_IT.pdf.

[ISO13] ISO 20022. Financial services - Universal financial message scheme. Part 1–8. International Organization for Standardization (ISO), 2013.

[JBC17] Jones, M., Bradley, J., Campbell, B.: OpenID Connect Token Bound Authentication 1.0 - draft 01. The OpenID Foundation, 2017. https://openid.net/specs/openid-connect-token-bound-authentication-1_0.html.

[JBS15a] Jones, M., Bradley, J., Sakimura, N.: RFC 7515. JSON Web Signature (JWS). Internet Engineering Task Force (IETF), 2015. https://tools.ietf.org/html/rfc7515.

[JBS15b] Jones, M., Bradley, J., Sakimura, N.: RFC 7519. JSON Web Token (JWT): Internet Engineering Task Force (IETF), 2015. https://tools.ietf.org/html/rfc7519.

[JCM15] Jones, M., Campbell, B., Mortimore, C.: RFC 7523. JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. Internet Engineering Task Force (IETF), 2015. https://tools.ietf.org/html/rfc7523.

[JH12] Jones, M., Hardt, D: RFC 6750. The OAuth 2.0 Authorization Framework: Bearer Token Usage. Internet Engineering Task Force (IETF), 2012. https://tools.ietf.org/html/rfc6750.

[KS10] Klingenstein, N., Scavo, T.: SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0. OASIS, 2010. https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-02.html.

[La00] Last, C. d., Gross, G., Gommans, L., Vollbrecht, J., Spence, D.: RFC 2903. Generic AAA Architecture. Internet Engineering Task Force (IETF), 2000. https://tools.ietf.org/html/rfc2903.

[LH09] Lockhart, H., Hardjono, T.: SAML V2.0 Condition for Delegation Restriction Version 1.0. Committee Specification 01. OASIS, 2009. https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cs-01.html.

[LM05] Linn, J., Mishra, P.: SSTC Response to "Security Analysis of the SAML Single Sign-on Browser/Artifact Profile". Working Draft 01. OASIS, 2005. https://www.oasis-open.org/committees/download.php/11191/sstc-gross-sec-analysis-response-01.pdf.

[Ma17] Mainka, C., Mladenov, V., Schwenk, J., Wich, T.: SoK: Single Sign-On Security–An Evaluation of OpenID Connect. 2017. IEEE European Symposium on Security and Privacy (EuroS&P), Conference Publishing Services (CPS), Paris, pp. 251-266, 2017.

[Me14] de Medeiros, B., Scurtescu, M., Tarjan, P., Jones, M.: OAuth 2.0 Multiple Response Type Encoding Practices. The OpenID Foundation, 2014. https://openid.net/specs/oauth-v2-multiple-response-types-1_0.html.

[MMS14] Mayer, A., Mladenov, V., Schwenk, J.: On the Security of Holder-of-Key Single Sign-On. Sicherheit, 2014, pp. 65-77.

[PW03] Pfitzmann, B., Waidner, M.: Analysis of liberty single-sign-on with enabled clients. IEEE Computing, 07/03, IEEE Computer Society, New Jersey, pp. 38-44, 2003.

[Po17] Popov, A., Nystroem, M., Balfanz, D., Langley, A., Hodges, J.: Token Binding over HTTP. draft-ietf-tokbind-https-09. Internet Engineering Task Force (IETF), 2017. https://tools.ietf.org/html/draft-ietf-tokbind-https-09.

[Re09] Regulation (EC) No 924/2009 of the European Parliament and of the Council of 16 September 2009 on cross-border payments in the Community and repealing Regulation (EC) No 2560/2001. Official Journal of the European Union, 2009. http://data.europa.eu/eli/reg/2009/924/oj.

[Re14] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Official Journal of the European Union, 2014. http://data.europa.eu/eli/reg/2014/910/oj.

[Re15] Regulation (EU) 2015/751 of the European Parliament and of the Council of 29 April 2015 on interchange fees for card-based payment transactions. Official Journal of the European Union, 2015. http://eur-lex.europa.eu/eli/reg/2015/751/oj.

[Re16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, 2016. http://data.europa.eu/eli/reg/2016/679/oj.

[Ri13] Rissanen, E.: eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard. OASIS, 2013. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.doc.

[Sa14a] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C.: OpenID Connect Core 1.0. The OpenID Foundation, 2014. https://openid.net/specs/openid-connect-core-1_0.html.

[Sa14b] Sakimura, N., Bradley, J., Jones, M., Jay, E: OpenID Connect Discovery 1.0. The OpenID Foundation, 2014. https://openid.net/specs/openid-connect-discovery-1_0.html.

[SBJ14] Sakimura, N., Bradley, J., Jones, M.: OpenID Connect Dynamic Client Registration 1.0. The OpenID Foundation, 2014. https://openid.net/specs/openid-connect-registration-1_0.html.

[SP03] Sandhu, R., Park, J.: Usage Control: A Vision for Next Generation Access Control. In V. Gorodetsky, L. Popyack, V. Skormin (Ed.). Computer Network Security 2776/03, Springer, Berlin, pp. 17-31, 2003.

[SLG10] Schwenk, J., Liao, L., Gajek, S.: Stronger bindings for SAML assertions and SAML artifacts. In SWS '08 Proceedings of the 2008 ACM workshop on Secure web services, Alexandria 2008, ACM Press, New York, pp. 11-20, 2008.

[So12] Somorovsky, J., Mayer, A., Schwenk, J., Kampmann, M., Jensen M.: On Breaking SAML: Be Whoever You Want to Be. In Proceedings of the 21st USENIX Security Symposium, Bellevue 2012, The USENIX Association, Berkeley, pp. 397-412, 2012.