Automated Risk Assessment and What-if Analysis of OpenID Connect and OAuth 2.0 Deployments

Salimeh Dashti, Amir Sharif, Roberto Carbone, Silvio Ranise

▶ **To cite this version:**

Salimeh Dashti, Amir Sharif, Roberto Carbone, Silvio Ranise. Automated Risk Assessment and What-if Analysis of OpenID Connect and OAuth 2.0 Deployments. 35th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), Jul 2021, Calgary, AB, Canada. pp.325-337, 10.1007/978-3-030-81242-3_19 . hal-03677036

HAL Id: hal-03677036

https://inria.hal.science/hal-03677036

Submitted on 24 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Automated Risk Assessment and What-if Analysis of OpenID Connect and OAuth 2.0 Deployments

Salimeh Dashti[1,2] [0000−0001−5595−2840], Amir Sharif[1,2] [0000−0001−6290−3588], Roberto Carbone[1] [0000−0003−2853−4269], and Silvio Ranise[1,3] [0000−0001−7269−9285]

[1] Fondazione Bruno Kessler, Trento, Italy
sdashti,asharif,carbone,ranise@fbk.eu
[2] DIBRIS, University of Genova, Genova, Italy
[3] Department of Mathematics, University of Trento, Trento, Italy

Abstract. The introduction of the Payment Service Directive (PSD2) has accelerated financial services and open banking growth. Deploying appropriate identity management solutions is crucial. This implies the adoption of secure protocols for authentication and authorization, such as OpenID Connect and OAuth 2.0. The PSD2 also requires the application of the General Data Protection Regulation (GDPR) when transactions involve personal data. In turn, the GDPR mandates a Data Protection Impact Assessment (DPIA) for assessing risks posed to data subjects' rights and freedom. This is a time-consuming and challenging task requiring heterogeneous skills that include the knowledge of best practices for deploying protocols, security mechanisms adopted by available identity management providers, and the capability to perform careful what-if analysis of the possible alternatives. To assist users in this task, we propose a methodology based on the formalization of the what-if analysis as an optimization problem that available tools can solve. The formalization is derived from the OAuth 2.0 and OpenID connects standards, security best practices to mitigate threats, and thorough the evaluation of 19 identity management providers to check their supported features concerning the identified set of features for OAuth/OIDC solutions. We apply the methodology to assist controllers and identify the most appropriate security setup to drive the process of making financial services compliant with the PSD2.

Keywords: Digital Identity · PSD2 · OAuth 2.0 · OIDC · GDPR · DPIA

1 Introduction

The growing importance of financial services and the regulatory push by the PSD2 to share user's financial data held by banks with third-party services have made the market more competitive. Although that brings a range of economic opportunities, it comes together with risks such as loss or theft of personal data, data protection violations, etc. One of the key points to make these services trustworthy—as required by PSD2—is to deploy an appropriate identity management solution. OAuth 2.0 (OAuth) and OpenID Connect (OIDC) are two widely used solutions among other identity management solutions. PSD2 states that where personal data is processed—as in the authorization and authentication process—relevant security requirements laid down in the General Data Protection Regulation (GDPR) should be met.

OAuth/OIDC solutions provide a secure and frictionless process [19,8,20]. However, wrong implementation choices of these solutions may result in data breaches that impact the rights and freedoms of data subjects in a large scale. A recent example is the internet bank account takeover of +1M users without user interaction due to an implementation flaw within their OAuth solution [7]. GDPR requires conducting Data Protection Impact Assessment (DPIA) to identify and evaluate risks to data subjects' rights and freedoms where the processing involves a large amount of personal data and affects a large number of data subjects (recital 75). That is true for OAuth/OIDC solutions.

Conducting a DPIA-compliant risk assessment for OAuth/OIDC solutions requires to: (1) assess the risk for both IdMP's deployment and its integration within web applications (hereafter clients); (2) face a maze of documents and guidelines to perform a comprehensive and flawless risk assessment which is a challenging task for non-security experts; (3) be aware of which Best Current Practices (BCPs) to follow that meets the requirements of their clients, e.g., for PSD2 open banking, they need to consider Financial-grade API (FAPI) [16] instead of OAuth/OIDC Core documents [4,17]; and (4) be aware of DPIA requirements to conduct a risk analysis. Meeting such requirements is a daunting task whose burden, according to the GDPR, is on the shoulder of the (data) controller. We propose a methodology to conduct a DPIA-compliant risk assessment for OAuth/OIDC solutions to assist controllers. It is designed to address the requirements of OAuth/OIDC-based financial services and any OAuth/OIDC-based client processing either a special category of personal data or common personal data. Our main contributions are:

- demystifying the maze of OAuth/OIDC documents to create a reference model that characterizes: (i) secure IdMPs deployment and their integration, and (ii) privacy-preserving components to meet DPIA requirements;
- formalizing a what-if DPIA-compliance risk analysis as an optimization problem, using the introduced reference model;
- proposing a methodology to solve the optimization problem and assist controllers in modeling and evaluating risks.

Paper Structure. The remainder of this paper is organized as follows. Section 2 introduces the background concepts used in the paper. Section 3 articulates the problem our methodology solves and our OAuth/OIDC security and privacy reference model. Section 4 presents our proposed methodology. Section 5 concludes the paper and provide some insights for future work.

2 Background

This section discusses the related concepts that make the paper self-contained.

Access Delegation: OAuth Standard. The OAuth authorization framework enables a third-party client to obtain access to a resource server, either on behalf of the resource owner or on its behalf [4]. The client redirects the resource owner through the browser into the IdMP authorization server, where the resource owner performs the authentication. After the successful authentication of the resource owner, the authorization server issues

an *access_token* which clients use to access the resource owners' resources in the resource server.

Single Sign-On Login: OIDC Standard. OIDC is an authentication layer developed on top of the OAuth standard, that adds two main features: `id_token` and *userInfo* endpoint. An `id_token` enables the client to verify that the received token is issued for its previous token request. It is a structured JSON token [6] that contains information about: token issuer, subject, and the audience (the intended client); all signed by the OIDC provider (IdP). An *userInfo* endpoint is to obtain identity-related attributes concerning the users (e.g., the email and address).

Privacy Goals. Security goals (confidentiality, integrity, availability, i.e. the CIA) need to be complemented with further privacy goals [3] to evaluate the impact on all aspects of privacy and data protection. They are: data unlinkability, data minimization, purpose specification, transparency, and intervenability. Recent research efforts have come up with these goals [14,1,13,3] to provide an interdisciplinary standard model to assess the consequences of a complex IT systems concerning privacy and data protection [3]. As the CIA are well-known, we discuss only the last three. *Data unlinkability* refers to hiding the link between two or more actions, identities, and pieces of information [22]. *Data minimization* requires avoiding unnecessary data to achieve the determined purpose, that is, *purpose determination*. The mentioned privacy goals are requested by article 6.4.e and 32.1.a. *Transparency* requires data processing to be understandable and reconstructable by concerned individual [1] (article 5.1.a and 12.2). *Intervenability* requires that intervention (for the individual whose data are processed) is possible concerning all ongoing or planned privacy-relevant data processing [1] (article 12.2). To comply with the GDPR, controllers need to address data processing principles (article 5). Beside, *accountability*, *data accuracy* and *storage limitation*, the rest overlap with the privacy goals.

Data Protection Impact Assessment. GDPR requires controllers to carry out a Data Protection Impact Assessment (DPIA). Among others, when the processing involves a large amount of personal data and affects many data subjects. The controller is the natural or legal person, public authority, agency, or other body that determines the purposes and means of processing personal data (article 4). DPIA is a risk-based approach to data protection. Article 35.7 articulates the steps to conduct it; which overlap with risk management steps (e.g., ISO 3100026).

3 Problem Definition and Reference Model

Analyzing risks to the rights and freedoms of data subjects is an important step of a DPIA. Conducting risk analysis for OAuth/OIDC solutions—that are the backbone of clients—is critical but complex. That is due to: many choices of IdMPs with various configuration options, various implementation patterns, many security implications and guidelines documents, and lack of guideline to meet privacy goals (see Section 2.3) using OAuth/OIDC components. Therefore, we propose a tool-based methodology to perform a DPIA-compliant risk analysis that requires: to consider the security and privacy level required by the client in question, e.g., for open banking, they need to consider FAPI instead of OAuth/OIDC Core documents; and to meet the privacy goals. As such, we identify the following problem:

P_{risk} *security and privacy risk assessment and what-if analysis, taking into account the risk propagation*.

Below, we introduce a reference model of OAuth/OIDC solutions (Section 3.1). We briefly describe how the usage of recommended components by the OAuth WG and OIDF can help to achieve the proper level of security and privacy (Section 3.2). Finally, we formalize the identified problem (Section 3.3).

3.1 OAuth/OIDC Reference Model

To build the reference model, we studied OAuth/OIDC security documents (e.g., [4,17]) to extract the components that protect OAuth/OIDC deployment, and satisfy privacy and security goals (introduced in Section 2.3). We call these components *atomic features*

Table 1: OAuth/OIDC Reference Model

Deployment place	Atomic features	Threat (T)	Goal (G)	Consecutive T	Consecutive G	PL	LL
		Security Feature					
Authorization request	code	Obtain AT	PD Conf	-	-	3	3
	token					1	
	client_credentials					3	
	password					1	
	hybrid					3	
	nonce	Obtain AT , Session misuse	PD Conf	-	-	5	5
	state	Session misuse	PD Conf	-	-	5	5
	request	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
	request_uri					5	
	query					1	
	form_post	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
	fragment					1	
	query					1	
	Code challenge	Obtain AT , Session misuse	PD Conf	-	-	5	5
	plain	Obtain AT , Session misuse	PD Conf	-	-	1	5
	s256					5	
Token request	mtls	Obtain AT	PD Conf	-	-	3	3
	client_secret_jwt					3	
	private_key_jwt					3	
	client_secret_basic					2	
	cleint_secret_post					2	
	code_verifier	Obtain AT , Session misuse	PD Conf	-	-	5	5
Authorization request / Token request	full redirect_uri[idmp]	Obtain code	AT Conf	Obtain AT	PD Conf	3	3
	pattern redirect_uri[idmp]					1	
Header	binding IdMP metadata[cl]	Obtain AT	PD Conf	-	-	2	2
	referrer[cl]	Obtain code	AT Conf	Obtain AT	PD Conf	3	3
Console	distinct redirect_uri[cl]	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
Client check	open redirect validation[cl]	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
	state validation[cl]	Obtain code	PD Conf	-	-	5	5
	storing IdMP metadata[cl]	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
	issuer validation[cl]	Obtain code	AT Conf	Obtain AT	PD Conf	5	5
	id token validation[cl]	Impersonation	PD Conf	-	-	5	5
		Privacy Features					
Authorization request	claims	Comp data mini	Data mini	-	-	5	5
	scope	Comp data mini	Data mini	-	-	5	5
	purpose	Comp. Pur spec, Trans	Pur Spec, Trans	-	-	5	5
	verified_claims	Comp. PD accuracy	PD accuracy	-	-	5	
	vot	Impersonation, Comp PD accuracy	PD accu, PD Conf	-	-	5	5
	acr					5	
	login	Comp transparency	Transparency	-	-	3	3
	select_account					3	
	consent					3	
Console	pairwise[cl]	Linkability	Unlinkability	-	-	5	5
	public[cl]					1	

LEGEND. idmp: idmp only feature, cl: client only feature, comp: compromise, PD: personal data, AT: access token, trans: transparency data mini: data minimization, conf: confidentiality, pur spec: purpose specification, PL: protection level, LL: likelihood level

(*af*), listed in Table 1. Each *af* represents either: (*i*) OAuth/OIDC parameter name (e.g., `nonce`), (*ii*) OAuth/OIDC parameter value (e.g., `code`), (*iii*) OAuth/OIDC functionality aspects (e.g., `full redirect_uri`), or (*iv*) client specific implementation-related tasks (e.g., `Id token validation`). A detailed definition for each *af* in the reported categories is provided in our companion website.[4] *af*s are to be set/implemented either: (*i*) in Authorization/Token requests; (*ii*) in the IdMP developer console, set by controllers; (*iii*) parameters set in the Header; or (*iv*) checks implemented by the controller. Table 1 represents that by column *Deployment Place*. For example, controller needs to set *subject identifier type*[5], that could be either `pairwise` or `public`, in the IdMP developer console. Note that, due to the page limit and simplifying the table, we only consider the *Authorization Code* flow, and interested readers can refer to our companion website[4] for the completed reference model.

The OAuth Working Group (WG) and the OpenID Foundation (OIDF) recommends different *af*s—with different levels of contributes (*protection level*)—to achieve a common goal for varied use-case scenarios. We group together such *af*s, and call them *composed feature* (*cf*). A client can implement one *af* from a *cf*, per request. For example, `request, request_uri` and `query` are grouped together under *cf Request*. They are introduced to meet goal *access token confidentiality*, by protecting authorization request against the threat *obtain code* (see Table 1). While *af* `request` and `request_uri` pass OAuth/OIDC requests in a signed and optionally encrypted single, self-contained parameter manner, the *af* `query` passes it directly in the URL. Thus, the first two provide a higher protection level as they provide request integrity and confidentiality. We have used this reasoning to assign protection level to the *af*s. Thus, the protection level for the discussed *af*s are 5, 5, and 1, respectively. This is an important consideration, as it allows controllers to make an informed decision on the IdMP s/he chooses (not all IdMPs support all the *af*s), or/and the *af*s they decide to implement. When an *af* is not implemented, the protection level against its related threat(s) will decrease and adds to the likelihood level of the threat(s) to pose. Such that, the likelihood level of *af*s are equal to their protection level. While in case of *cf*s, the likelihood level will be the maximum value among its *af*s Protection/likelihood levels range from 1 to 5. Controllers can modify the likelihood and protection level if needed. Please find the details about the evaluation of the protection level of all *af*s in our website.[4]

Each *af*/*cf* is introduced to protect OAuth/OIDC deployment against a threat to satisfy privacy/security goals. A threat could expose the system to another threat, which itself compromises another goal. We call them *consecutive threats* and *consecutive goal*. They are as likely to raise as their main threat. For example, as represented in Table 1, not implementing *af* `request` leads to threat *obtain code*, which itself let the attacker to *obtain access token*. They relate to goal *access token confidentiality*, and *personal data confidentiality*, respectively.

[4] `https://sites.google.com/fbk.eu/oidc-dpia`

[5] A locally unique and never reassigned identifier within the Issuer for the user, which is intended to be consumed by the Client

3.2 Best Current Practice Specification

To assist clients and IdMPs in achieving appropriate security and privacy levels based on their operating domain, the OAuth WG and the OIDF have published a set of BCPs in [9,16,10,17,4]. Depending on the domain the BCPs mandate to use some optional *af*s; and for *cf* to use an *af* over the others. For example, [16] requires using optional *af* `state`.

The OAuth WG and the OIDF do not provide any specific privacy considerations to meet privacy goals (See Section 2.3). However, to comply with the DPIA requirements, controllers need to address them. We have systematically studied the following documents [5,21,12] to provide easy-to-implement privacy recommendations for controllers based on the reported *af*s in Table 1. For the sake of brevity, we omit the full explanation and only give a couple of examples. As reported in Table 2 *cf* Response type comprises of *af*s `code, token, client, credentials, password,` and `hybrid`. However, the BCPs enforces the usage of *af* `code` among the other *af*s because it does not return the *access_token* in the front channel and it can be protected by *af*s `code challenge` and `code_verifier`. Concerning privacy, controllers can achieve privacy goal *purpose specification* by using *af* `purpose` to state the purpose of asking each individual claim.

3.3 Problem P_{risk} definition

In this section, we formalise the problem, namely P_{risk}, considering the reference model reported in Table 1.

Let $\mathcal{AF}$ be the set of *af*s associated with an OAuth/OIDC deployment shown in the second column of Table 1. We split the set $\mathcal{AF}$ in three disjoint subsets: $\mathcal{AF} = \mathcal{AF}_{\text{common}} \cup \mathcal{AF}_{\text{idmp}} \cup \mathcal{AF}_{\text{cl}}$. The set $\mathcal{AF}_{\text{common}}$ includes the *af*s that the client cannot implement unless the IdMP supports them, like `nonce`. An IdMP can support more than one *af* from a *cf*, while the client can implement only one for a given request. The set $\mathcal{AF}_{\text{idmp}}$ includes *af*s that IdMPs need to enforce (marked with the "idmp" superscript in Table 1) and clients need only to adopt, like `pattern redirect_uri`. The set $\mathcal{AF}_{\text{cl}}$ includes the *af*s that clients need to implement and checks that they need to perform (marked with the "cl" superscript in Table 1), like `issuer validation`.

Let $support_{idmp}$ be a Boolean mapping from the set $\mathcal{AF}_{\text{common}} \cup \mathcal{AF}_{\text{idmp}}$ for *idmp* ranging over a given set of IdMPs. An *af* in $support_{idmp}$ maps to a true value *iff idmp*

Table 2: Composed Features.

Composed Feature	Related Atomic Features
Response type	`{code,token,client_credentials,password,hybrid}`
Request	`{request,request_uri,query}`
Response mode	`{form_post,fragment,query}`
Code challenge method	`{plain,SHA256}`
Client authentication	`{mTLS,client_secret_jwt,private_key_jwt,client_secret_basic,client_secret_post}`
Redirect uri	`{full redirect_uri,pattern redirect_uri}`
Identity proofing	`{vot,acr}`
Prompt	`{login,select_account,consent}`
Subject type	`{public,pairwise}`

supports the *af*. Let $implement_{cl}$ be a Boolean mapping from the set $\mathcal{AF}_{\text{common}} \cup \mathcal{AF}_{\text{cl}}$ for a given client *cl*. An atomic feature *af* maps to a true value *iff cl* implements *af*.

Notice, for $af \in \mathcal{AF}_{\text{common}}$, we say that $implement_{cl}$ is constrained by $support_{idmp}$, that is, $implement_{cl}$ is strictly dependent on $support_{idmp}$. Other words, $af \in \mathcal{AF}_{\text{common}}$ can map to true only if it maps to true in $support_{idmp}$. Indeed, controllers can decide whether to implement some atomic features among the one supported by *idmp*.

Let $\mathcal{CF}^* \subset \mathbf{P}(\mathcal{AF})$ (where **P** stands for the power set) be a set of sets of *af*s, including the *cf*s (see Table 2) as well as a set $\{af\}$ for each atomic feature $af \in \mathcal{AF}$ that does not belong to any composed feature in Table 2. Thus, $\mathcal{CF}^* = \{\ldots, \{\texttt{nonce}\}, \{\texttt{state}\}, \{\texttt{request}, \texttt{request\_uri}, \texttt{query}\}, \ldots\}$. Note that all the sets $S \in \mathcal{CF}^*$ are pairwise disjoint and $\bigcup_{S \in \mathcal{CF}^*} S = \mathcal{AF}$, that is $\mathcal{CF}^*$ is a partition of $\mathcal{AF}$. Let $\mathcal{T}$ be the set of threats and consecutive threats, listed in columns 3 and 5 of Table 1. Let $\mathcal{G}$ be the set of goals and consecutive goals, listed in columns 4 and 6 of Table 1. We thus define the following mappings and relations:

- Let p be a mapping from $\mathcal{AF}$ to the set $\{1, \ldots, 5\}$ of protection levels. The definition of this mapping can be obtained by considering the features in column 2 and the corresponding protection level in column 7 of Table 1.
- Let the likelihood mapping ℓ be a mapping from $\mathcal{CF}^*$ to the set $\{1, \ldots, 5\}$ of likelihood levels. The definition of this mapping can be obtained by considering the (sets of) atomic features in column 2 and the corresponding likelihood level in column 8 of Table 1.
- Let i be a mapping from the set $\mathcal{T}$ of threats to the set $\{1, \ldots, 5\}$ of impact levels. The definition of this mapping is decided by the controller and depends on the particular scenario in which the OAuth/OIDC solution is deployed.
- Let $CF2T \subseteq \mathcal{CF}^* \times \mathcal{T}$ be a relation between each composed feature $cf \in \mathcal{CF}^*$ and its related threat. The pairs in this relation can be defined by reading the elements reported in columns 2 and 3 of Table 1.
- Let $T2G \subseteq \mathcal{T} \times \mathcal{G}$ be a relation between each threat and the goal compromised by the threat itself. The pairs in this relation can be defined by reading the elements reported both in columns 3 and 4, and 5 and 6 of Table 1. Indeed, the relation between a threat and the goal is independent of the fact that the threat is consecutive to another threat or not.
- Let $T2CT \subseteq \mathcal{T} \times \mathcal{T}$ be a relation between a threat and its consecutive threat. The pairs in this relation can be defined by reading the elements reported in columns 3 and 5 of Table 1. We also use the notation $CT(t)$ to indicate the set of threats consecutive to the threat t, and $CT(T) \triangleq \bigcup_{t \in T}(CT(t))$.

The problem P_{risk} consists in finding the *idmp* and mapping $implement_{cl}$ such that

$$min_{idmp, implement_{cl}} \mathcal{R}(p, \ell, i, support_{idmp}, implement_{cl}, CF2T, T2G, T2CT) \qquad (1)$$

subject to $idmp \in$ IdMPs and $cl \in ClAdm$, where IdMPs is a set of IdMPs that support certain features and *ClAdm* is the set of admissible mappings for a given client. The definition of the set IdMPs and the *ClAdm* derive from external considerations performed by the controller of the client. For instance, the choice of an IdMP can be affected by the

costs of IdMPs' services or supported features. Similarly, the controller can consider that some features—among the ones constrained (as defined above) by the selected *idmp*—will take longer to implement or charge more in terms of costs. Thus, the controller can further constrain the admissible $implement_{cl}$ mappings accordingly.

$\mathcal{R}$ is thus an operator that returns the risk level given the selected *idmp*, the configuration of the client *cl*, the protection and likelihood mappings and the impact level while considering how the risk propagates among the various components by using relations *CF2T*, *T2G*, and *T2CT*.

As a final remark, note that the problem P_{risk} is solved by considering *p*, *l*, *i*, *CF2T*, *T2G*, and *T2CT* (obtained from Table 1), while $support_{idmp}$ and $implement_{cl}$ range over all possible values in the sets IdMPs and *ClAdm*. Therefore, the problem is decidable because it can be expressed as a combinatorial optimization problem with finitely many possibilities depending on the number of considered IdMPs, features supported by clients in *ClAdm* and on the cardinality of the considered atomic features in $\mathcal{AF}$: all the possibilities can be enumerated, the corresponding risk evaluated and the minimum value selected. Indeed, solving problem P_{risk} can be mechanized by using any available tools that are capable of solving optimization problems by specifying how the function to be minimized (the risk in our case) depends on the arguments of the operator $\mathcal{R}$.

4 OAuth/OIDC Solution DPIA-Compliant Risk Analysis

This section discusses our methodology to address the reported problem.
Running Example. *Graphy* is a client that gets the users' financial data from their bank accounts and makes a graphical representation of their financial status. Users need to make a profile with the *Graphy* and connect their bank accounts to it. *Graphy* utilizes OAuth/OIDC solutions to provide a smooth single sign-on login and access delegation experience for its users. This scenario is inspired by the example provided in [15].

Addressing P_{risk}**.** The problem assesses the risks to rights and freedoms of data subjects, for which controllers need to meet security and privacy goals (introduced in Section 2.3). This section details our three-step approach to address the problem, namely: *(I) assessing client, (II) evaluating risks of employed IdMPs*, and (III) *modeling and treating risks*.
I. Assessing Client. This sub-step identifies the roles; namely data subject, controller, and data processor; and data type category. Controller decides on data processing details and is responsible to demonstrate compliance with GDPR; data processor processes personal data on behalf of the controller. In this context, the client developer is the controller and, IdMPs are the data processor. We provide three ways for controllers to identify the employed IdMPs: finding the employed IdMP from list of popular IdMPs, indicating discovery endpoint URL, replying to a questionnaire. To identify data subjects and data type, we use the approach introduced in [2], that is, through specifying the economic sector. We call *Sensitive sector* the *sectors* that process special category of personal data and/or involve vulnerable data subject; and *non-sensitive sector* otherwise.

DPIA considers the impact of data processing *high*, when a large scale of data is involved, which impacts large scale of the data subject. For the sake of simplicity, we consider the impact level as a constant value 5 corresponding to a very high impact.

Application of the step on the scenario. In this scenario, Graphy is the controller. The service it provides belongs to *Sensitive* sector as it processes financial data. As such, the impact is 5. Data subjects are natural persons. The controller employs *IBM* for single sign-on login and allows users of *Barclays* bank to link their bank accounts through access delegation; they are data processors.

II. Evaluating Risks of Employed IdMP(s). This sub-step assesses employed IdMPs and the implementation details w.r.t. their integration within the client, to identify any threat(s) they may pose to the right and freedoms of data subjects. For that, we introduce the following two components: *IdMP Processor* and *Client Processor*, detailed below.

IdMP Processor. The processor assesses which are the features $af \in \mathcal{AF}_{\text{common}} \cup \mathcal{AF}_{\text{idmp}}$ supported by the employed IdMP(s). Thus, more formally, the IdMP processor aims to specify the $support_{idmp}$ Boolean mapping for the employed *idmp*.

Our methodology provides a database of known IdMPs. If the controller selects one of them, the known truth values are automatically retrieved. Otherwise, we identify two solutions to collect the missing information about the *af*s of an *idmp*. One option for the controller is to provide the IdMP's discovery endpoint URL (if available). Then, our tool sends an HTTP Get request to the endpoint and automatically extracts the information from the JSON response (e.g., `subject_type_supported`). (Please refer to the companion website[4] for the list of the *af*s included in the JSON response). Finally, we ask the controller to reply to a dynamically generated Yes/No questionnaire to collect only the information that is still missing. To generate the questionnaire, we use the reference model (see Section 3.1), by filtering out the known information. For example, it asks "Does the IdMP support `full redirect_uri`?", "Does the IdMP support `nonce`?". Please find the details of the questionnaire in our website.[4]

The database of known IdMPs gets updated whenever a new IdMP is introduced either via the discovery endpoint or the questionnaire. To know the status of popular IdMPs and to start filling in the database, we have selected 19 popular IdMPs, according to their Alexa ranks and whether they have a free developer console to assess their features. The IdMPs are taken from the OIDF website [11]: for *Sensitive sector* they are taken from the list of *Certified FAPI OIDC*, and for *Non-sensitive sectors* from *certified OIDC Core*. For each of them we studied the available documentation and/or the developer console to assess their features. Table 3 provides an excerpt of the IdMPs we have studied; the full table is available in our website.[4] Interesting enough, we discovered that the selected IdMPs do not provide the same level of security and privacy. For instance, *Yahoo* does not support the following *af*s: `Plain, S256, claims, purpose, acr, verified_claims` and `vot`, while the *IBM solution* supports `Plain, S256` and `acr`.

Client Processor. The Client Processor aims to assess which features $af \in \mathcal{AF}_{\text{common}} \cup \mathcal{AF}_{\text{cl}}$ have been implemented by the client *cl* in the OAuth/OIDC solutions. Other words, it specifies the $implement_{cl}$ boolean mapping.

To support the controller in this process, our methodology provides a Yes/No questionnaire. By leveraging the $support_{idmp}$ (obtained from the IdMP Processor) our approach automatically takes into account the choices that are constrained (as defined in Section 3.3) by the employed *idmp* and selects the relevant questions for the client. For example, it asks "*Have you implemented* `id token validation`*?*", "*Have you set* `referrer` *parameter?*".

Application of the step on the scenario. Given that *IBM* is in our database of known IdMPs then most of truth values of the features related to *IBM* are already available. For the missing information concerning *IBM* and the whole information concerning *Barclays*—that is not among the IdMP we analyzed—the controller has to reply to the questionnaire. The details about the collected information are on the website.[4]

III. Modeling and Treating Risks. The information collected from the previous two steps in Section 4.I and Section 4.II (namely the impact i, $support_{idmp}$, and $implement_{cl}$) as well as the information reported in Table 1 are used to model and evaluate the risk. Any risk modeling tool that can perform a what-if analysis can be employed. The what-if analysis allows controllers to observe how the risk level changes by considering different implementation choices. A good option is the RiskML tool [18], which provides a modeling language and a quantitative reasoning algorithm to analyze models, taking into account the risk propagation.

Figure 1 illustrates an excerpt of the risk model, by considering, for simplicity, only two *af*s: `referrer` and `mtls`. We are assuming that, according to $implement_{cl}$, a controller did not implement `referrer` (dashed line in Figure 1) and implemented `mtls`. As shown in Table 1, not implementing `referrer` leads to the threat *Obtain Code* (with the likelihood level 3), that consecutively allows an attacker to also *Obtain Access Token*. As such, it impacts the goals *confidentiality of the access token* and *confidentiality of the personal data*, respectively. At the same time, `mtls` is implemented, and thus

Atomic Feature	IBM	Google	box	Yahoo
`code`	✓	✓	✓	✓
`implicit`	✓	✓	✗	✓
`client credential`	✓	✗	✗	✗
`password`	✓	✗	✗	✗
`hybrid`	✓	✗	✗	✗
`mtls`	✓	✗	✗	✗
`client_secret_basic`	✓	✓	✓	✓
`client_secret_post`	✓	✓	✓	✓
`client_secret_jwt`	✗	✗	✗	✗
`private_key_jwt`	✓	✗	✗	✗
`plain`	✓opt	✓opt	✗	✗
`S256`	✓opt	✓opt	✗	✗
`request`	✓	✗	✗	✗
`request_uri`	✓	✗	✗	✗
`query`	✓	✓	✓	✓
`claim`	✓	✗	✗	✗
`scope`	✓	✓	✓	✓
`purpose`	✗	✗	✗	✗
`verfied_claims`	✗	✗	✗	✗
`acr`	✓	✗	✗	✗
`vot`	✗	✗	✗	✗
`public`	✓	✓	NA	✓
`pairwise`	✗	✗	NA	✗
Sensitive Sector	*No*	*No*	*No*	*No*

LEGEND: ✓opt: Supported, but not enforced, NA: Not applicable

Table 3: An excerpt list of known IdMPs and their supported features.

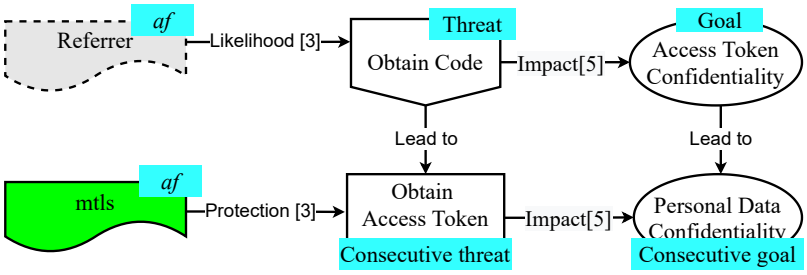

Fig. 1: An Excerpt Representation of Risk Model.

contributes to protect the *access_token* (the protection level is 3), and, as a consequence, the *confidentiality of the personal data*.

This simple excerpt shows that the effects of the implementation choices (expressed in terms of *af*s) propagate on threats and goals, and that the positive (protection) and negative (likelihood) effects contribute to the final risk level associated to each goal. Then, the specific operations to quantify the effects and calculate the risk levels are dependent on the employed risk assessment tool. Of course, when considering the whole set of *af*s the analysis is much more complex. By enabling controllers to perform a what-if analysis, our methodology allows them to take informed decisions on their IdMP and implementation choices.

5 Conclusion

Conducting a DPIA-compliant risk analysis for OAuth/OIDC solutions is complex. To assist controller with this task, we define a DPIA-compliant risk analysis as a security and privacy risk analysis (P_{risk}) problem, and proposed a methodology to solve it. The methodology is supported by a reference model that captures the OAuth/OIDC features that are required to solve the problem, respecting the security and privacy level of solution in question. Our analysis of the solution outputs a risk model that captures the system as-is, and provide the possibility to perform what-if analysis. Performing what-if analysis enables controllers to make an informed decision on their choice of IdMP and implementation to eliminate identified risks or minimize their impact. The model can be input into any risk model that can perform what-if analysis. As future work, we plan to extend the methodology to assess risks posed by (other components of) clients and introduce security controls and privacy-enhanced technology to address them.

References

1. Danezis, G., Domingo-Ferrer, J., Hansen, M., Hoepman, J.H., Metayer, D.L., Tirtea, R., Schiffner, S.: Privacy and data protection by design-from policy to engineering. arXiv preprint arXiv:1501.03726 (2015)
2. Dashti, S., Ranise, S.: A tool-assisted methodology for the data protection impact assessment. in proceedings of the international conference on security and cryptography (2019)

3. Hansen, M., Jensen, M., Rost, M.: Protection goals for privacy engineering. In: IEEE SPW (2015)
4. Hardt, D.: The oauth 2.0 authorization framework. IETF (2012)
5. Internet-Draft: International Government Assurance Profile (iGov) for OpenID Connect 1.0 (2018)
6. Jones, M., Bradley, J., Sakimura, N.: Json web token (jwt). IETF (2015)
7. Krebs, B.: Internet bank account takeover of +1m users without user interaction. https://mrbriankrebs.medium.com/internet-bank-account-takeover-of-1m-users-without-user-interaction-4fc9141740a3, accessed: 2021-03-25
8. Li, W., Mitchell, C.J.: User access privacy in oauth 2.0 and openid connect. In: EuroS&PW. IEEE (2020)
9. Lodderstedt, T., Bradley, J., Labunets, A., Fett, D.: OAuth 2.0 Security Best Current Practice (draft-ietf-oauth-security-topics-16). IETF (2020)
10. Lodderstedt, T., McGloin, M., Hunt, P.: Rfc 6819: Oauth 2.0 threat model and security considerations. IETF (2013)
11. OpenID Foundation: Financial-grade API - part 1: Baseline security profile. https://openid.net/certification/, accessed: 2020-23-11
12. Richer, J., Johansson, L.: Vector of trust (rfc 8485). IETF (2018)
13. Rost, M., Bock, K.: Privacy by design and the new protection goals. DuD **2009** (2011)
14. Rost, M., Pfitzmann, A.: Datenschutz-schutzziele—revisited. Datenschutz und Datensicherheit-DuD **33**(6), 353–358 (2009)
15. Sakimura, N.: Authorization delegation: A financial accounts aggregation use case. https://nat.sakimura.org/2016/01/29/authorization-delegation-a-financial-accounts-aggregation-use-case/, accessed: 2021-03-25
16. Sakimura, N., Bradley, J., Jay, E.: Financial-grade api - part 1: Baseline security profile Accessed: 2020-23-11
17. Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., Mortimore, C.: OpenID connect core 1.0 incorporating errata set 1. The OpenID Foundation **335** (2014)
18. Siena, A., Morandini, M., Susi, A.: Modelling risks in open source software component selection. In: International Conference on Conceptual Modeling. Springer (2014)
19. Similartech.com: Login providers. https://www.similartech.com/categories/login-provider, accessed: 2020-29-12
20. Sun, S.T., Beznosov, K.: The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In: Proceedings of ACM ASIACCS (2012)
21. Torsten, L., Daniel, F.: Openid connect for identity assurance 1.0. https://openid.net/specs/openid-connect-4-identity-assurance-1_0.html, accessed: 2019-19-06
22. Wuyts, K., Scandariato, R., Joosen, W., Deng, M., Preneel, B.: Linddun: a privacy threat analysis framework (2019)