On the (Privacy) Harms of the European Digital Identity Framework

Christian Knabenhans*
christian.knabenhans@epfl.ch
EPFL

Shannon Veitch*
shannon.veitch@inf.ethz.ch
ETH Zurich

Mathilde Raynal
mathilde.raynal@epfl.ch
EPFL

Theresa Stadler
theresa.stadler@epfl.ch
Swiss Data Science Center

Sylvain Chatel
sylvain.chatel@cispa.de
CISPA Helmholtz Center for Information Security

Wouter Lueks
lueks@cispa.de
CISPA Helmholtz Center for Information Security

Carmela Troncoso
carmela.troncoso@mpi-sp.org
EPFL, Max Planck Institute for Security and Privacy

Abstract

As digital identity frameworks (DIFs) gain traction around the world, many see privacy-enhancing technologies (PETs) as the key to prevent their potential negative societal impacts such as discrimination or surveillance. We critically examine whether PETs can achieve this goal using the European Digital Identity Framework (EUDIF) as an example. We develop a harm analysis methodology based on harm trees that illustrates how information leakage, certain design decisions, or a DIF's deployment context lead to harms. We leverage techniques from cryptographic modeling to formally capture the leakage of the core functionality of the EUDIF and its proposed applications. Our harm modeling elucidates which pathways to harm are inherent to the EUDIF's core functionality, and which pathways can be mitigated through PETs. Our analysis shows that, while PETs can reduce information flows, they fall short in actually mitigating the harms that deploying digital identity can bring to individuals and society.

1 Introduction

Digital identity frameworks (DIFs) equip users with digital credentials that authenticate their attributes, such as name, date of birth, gender, or education information. Users store credentials in a digital wallet, and can use them to access services (e.g., prove that they are older than 18) or demonstrate authenticity of documents (e.g., driving licenses), both online and in the physical world. DIFs are becoming prevalent across many jurisdictions [3, 54, 55, 60], with a prominent example being the *European Digital Identity Framework* (EUDIF), introduced by the European Union in April 2024 and poised for initial roll-out in late 2026. The EUDIF promises to provide 450 million Europeans with a "self-determined personal digital wallet" [22, Art. 5] that will streamline access to public and private services and increase trustworthiness in online interactions [13]. While the broad deployment of DIFs such as the EUDIF is anticipated to bring many benefits, civil society has raised concerns about its potential negative impact on certain user groups, for example through the exacerbation of structural exclusion, discrimination, or large-scale surveillance [21, 53, 66]. The EUDIF proponents' main response to these concerns has been to strictly follow "privacy-by-design" principles and to integrate privacy-enhancing technologies (PETs) into the system's design [13, 32, 34]. For example, following feedback from cryptographers [9], the EUDIF implementation recommendations include the use of anonymous credentials. Such privacy safeguards are portrayed as a sufficient step to mitigate harms that might arise from the EUDIF's deployment. In this paper, we ask:

To what extent are PETs an appropriate answer to concerns about the negative societal impacts of the EUDIF?

We make the following contributions.

New method for harm analysis. We introduce *a new approach to analyse harms* based on harm trees [18, 19], formal system modeling, and adversarial capability games. Together, these components allow us to identify pathways to, and root causes of, harm in any system (Sec. 3). Our formal model allows us to identify system components that are inherent to the intended functionality of a system, mimicking a perfectly privacy-preserving system with minimal information leakage. Alongside our model, we use a novel application of cryptographic games to demonstrate that the core functionality inherently enables certain adversarial capabilities that lead to harms. Our approach to harm analysis allows us to disambiguate which pathways to harms stem from (i) the core functionality of a system (e.g., intended system outputs, desired properties), (ii) deploying in a digital context, or (iii) a concrete implementation (including those integrating PETs).

Harms of the EUDIF. Using this methodology, we derive a broad range of (inherent) harms to individuals and society that might arise from the EUDIF, and make the following three additional contributions.

Formal model of EUDIF-like systems. We derive a formal model of the core functionality of the EUDIF from current regulations and specifications (Sec. 4). The model represents the best-possible privacy-by-design implementation of the EUDIF that reveals *only the minimal information* needed to fulfil its intended purpose. It captures all use cases suggested by the European Commission [27].

*These authors contributed equally to this work.

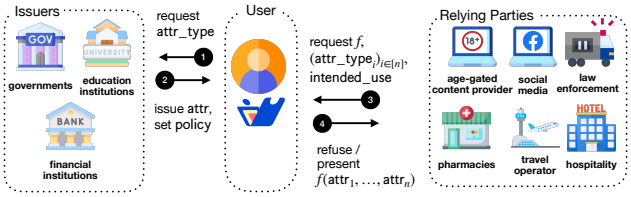


Figure 1: Overview of the EUDIF

Using credential-based age verification (Sec. 5.1) and identity authentication with mobile driving licenses (Sec. 5.2) as examples, we pinpoint components of the EUDIF that may lead to harms. These use case-specific models are based on international standards [41] and capture systems in jurisdictions beyond the EU, such as US-based mobile driving licenses.

Fundamental harm-enabling capabilities. We show that the core functionality of general-purpose DIFs such as the EUDIF enables three adversarial capabilities that lead to harms: distinguishing between users, tracking users, and inference of attributes (Sec. 6). Because these capabilities are inherent to the DIF functionality, they, and the harms that they enable, cannot be eliminated through technical means. Thus, only *constraints on the functionality*, enforced by regulations, can prevent adversaries from gaining these capabilities.

The limits of PETs. Our analysis demonstrates that, for every harm that we consider, there is always a pathway to this harm rooted in the core EUDIF functionality or in its digital nature. That is, even a perfectly privacy-preserving implementation of the EUDIF—one that leaks the minimal information to achieve the use case functionality—cannot prevent the harms that we analyse. PETs can only limit additional pathways to harm that arise in concrete implementations of the EUDIF, i.e., pathways that arise from additional information flows and parties not present in the abstract EUDIF functionality (Sec. 7). Finally, we reflect on our findings and discuss how some harms can only be mitigated by policy changes that heavily restrict the scope and functionality of DIFs (Sec. 8).

2 The European Digital Identity Framework

The eIDAS 2.0 regulation [40] establishes a European Digital Identity Framework (EUDIF), which mandates member states to have interoperable electronic ID systems, and opens the door for private sector actors to issue and verify attributes within this ecosystem. Fig. 1 gives an overview of the EUDIF: users have a digital wallet (e.g., a mobile application) that stores certified attributes from issuers. The wallet enables users to use these attributes to interact with service providers (called *relying parties*, or RPs) in a trustworthy manner.

Issuance. Users can request attributes to be issued to them (❶) by issuers like government agencies and public and private organizations. These attributes can be, e.g., about users' identity, membership in an organization, or education. Issuers issue authenticated attributes to the user's wallet (❷). The issuer may check that the user is entitled to the attribute (e.g., by verifying their identity), and may specify a disclosure policy for the issued attribute. This policy establishes the conditions under which the attribute may be disclosed to RPs.

Presentation. RPs can request users to present (functions of) certified attributes stored in their wallet (❸). This may be to authenticate the user (e.g., when requesting an official document), for access control (e.g., to perform age verification), or to verify information (e.g., law enforcement asking for the user's driver license at a traffic stop). The user can then approve or deny the RP's request. If approved, the wallet sends the requested (function of) attributes to the RP (❹).

Use cases. The European Commission documents use cases of the EUDIF [27] to illustrate the range of intended uses; we summarize these in App. C, Table 1. Several applications enable identification (i.e., full name, date of birth, nationality, and other details) of an individual in online or offline scenarios, such as for digital replacements for physical IDs, loyalty cards, and employment badges. The remaining use cases provide users with access to services, physical spaces, or financial advantages. These mainly belong to two extremes: those that require full identity information (e.g., mobile driving license, digital travel credential, ePrescription); and those that reveal one bit of information indicating whether a request should be authorized (e.g., age verification, European parking/disability card). In this work, we consider two use cases that are representative of these extremes (mobile driving license and age verification), and that are also considered for deployment in jurisdictions and contexts beyond the EU [1, 2, 6, 58].

3 A Methodology for Harm Analysis

In this section, we introduce a novel harm analysis methodology to analyse harms in digital systems and their origins. The methodology uses *formal modeling* and *harm trees* to identify the minimal leakage of a system that is necessary to achieve its functionality and to determine whether this minimal leakage is sufficient to provide the adversary with enough capabilities to materialize harms. It further enables us to analyse which pathways to harms can be prevented through PETs and other protective measures, and which harms are inherent to a system's core functionality.

3.1 Building Blocks

The methodology has three main building blocks: a formal model of a system's functionality, harm trees, and a novel application of security games to model adversarial capabilities.

Formal system model. To identify system components that may lead to harms, we formalize a system as an *ideal functionality*. This functionality abstracts concrete cryptographic mechanisms and specifies parties, their interfaces, information flows, and enforced constraints. The use of the syntax of (ideal) functionalities and the scope of our model is similar to Canetti et al.'s [11] approach to specifying the broader system and its properties. In our work, rather than for the usual purpose of proving security, we use an ideal functionality to identify origins of harms and to study PETs' effectiveness in preventing them. Towards this goal, the ideal functionality thus *represents a perfect privacy-preserving system* that leaks the minimal amount of information required to achieve the intended functionality.

Harm trees. We introduce harm trees, inspired by previous work that used trees for threat modeling [20, 64] and to identify privacy harms that arise due to access to information [17–19]. In a harm tree, the leaves represent *system components*, the intermediate layer represents *capabilities* that the adversary gains by accessing these components, and the root node represents a high-level *harm*. Compared to previous work, our harm trees can accommodate a much broader set of harms than those caused by information leaks. Also, the trees are designed such that paths in the tree serve to identify the origin of a harm beyond its mere existence. This, in turn, enables a precise analysis of the role of PETs in preventing pathways to harm (and thus the harm itself). While we apply harm trees to the analysis of the EUDIF, harm trees are broadly relevant to conduct harm-based analyses of any technological system.

Harms. We adopt the definition of harm as "an impairment or disruption of an entity's capacity or ability to function and exist as it otherwise would have in its usual context" [15]. Following the taxonomy of privacy harms of Citron and Solove [12] and the CyberPeace Institute's harms methodology [15], we consider as impairments any negative impact on individuals, groups of individuals, or the society as a whole, from the standpoint of physical, mental, or financial well-being or reputation, dignity, freedom, acceptance in society, self-actualization, domestic life, freedom of expression, or any other fundamental right (see App. A for details and examples).

Capabilities. Capabilities (or combinations thereof) can lead to harms. Examples include the capability to track users, the incapability of a provider to provide a service in a region, or the incapability of a user to access a service.

System components. A set of system components can enable certain *capabilities* (or lack thereof) on the system or its entities. We identify *system components* that might ultimately lead to harms. These comprise information leakage due to the use of the system, the results of an entity's interaction with the system, and requirements imposed on the entities by the system. For example, in the age verification use case of the EUDIF, system components include the leakage of information about a user's age to the RP, the decision of a user to engage with the age verification system, or the requirement to have an EUDIF-enabled device to interact with the system.

Adversarial capability games. We use a novel application of *security games* to capture the capabilities of an adversary given the (ideal) functionality of the system. To determine whether a functionality satisfies a given property, we use *universally-composable-compatible security games* [7]. Rather than employing security games to capture standard cryptographic properties, we use each security game to capture a *capability* that enables harm(s). If an adversary (e.g., malicious RP or issuer) can win the game, then we conclude that any implementation that realizes the system functionality also enables the corresponding capability. These security games allow us to precisely capture which system components can be exploited to enact harm.

3.2 Harm-Centered System Analysis

We combine these building blocks to analyse how digital systems lead to harms, and to determine when a protection is suitable to reduce or eliminate these harms.

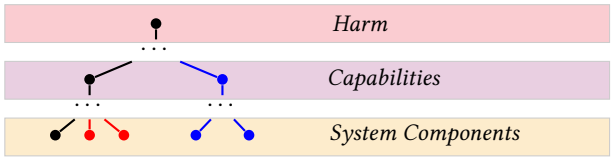


Figure 2: Harm tree. Components in black originate in the functionality, components in blue originate in the digital context of a use case, and components in red appear due to implementation choices.

Identifying pathways to harm. We first extract system components from the formal system model. Since this model represents the system's minimal information leakage, this step identifies leakage, requirements, or interactions in the core functionality (**functionality components**) that might cause harm.

While the formal model is designed to only reveal the information necessary to realize the functionality, in reality, an implementation might leak additional information. We also identify components that are inherent to the system being used in a *digital context* (**digital-context components**).

Finally, to study the effectiveness of protections, we consider components that would arise in a (non-perfect) concrete implementation of the system (**implementation components**). For the case of the EUDIF, we model components of a naive, concrete implementation where users send their complete attributes (e.g., a copy of their identity documents) to the RP upon each presentation, and also components that arise from deployment requirements of systems.

Building harm trees. When system components enable capabilities that result in harms, we say that we have found a *pathway* to harm. We combine these pathways to construct a *harm tree* that captures ways in which a harm may be caused: from system components, that (combine in some way to) enable capabilities, that (combine in some way to) enable harms (see Fig. 2). When representing trees, we use AND and OR operators to indicate how system components combine to create capabilities. Whenever more than one system component might be needed for the capability, we write OR^+.

Ascribing harm origins. Given a harm tree, we identify the root cause of each pathway to harm, according to the origin of its system components (i.e., whether they are **functionality**, **digital-context**, or **implementation** components). The harm trees enable the identification of additional types of components; for the purposes of our analysis, we focus only on these three here. **Functionality components** (and thus the harms that they enable) *cannot be eliminated* without preventing the system from operating as intended. Similarly, **digital-context components** *cannot be eliminated* because they cannot be eliminated from a *digital* system. On the other hand, **implementation components** *may be eliminated*, e.g., via the use of PETs or alternative implementation choices. From these observations, we draw conclusions about the extent to which harms in a system may be prevented by PETs, and which are inherent to the system design and its deployment context.

Fundamental capabilities. After identifying pathways to harm and their origins, we leverage our adversarial capability games, in

combination with the formal model, to understand the impact of **functionality components**, and which adversarial capabilities are enabled by the core functionality of the (perfectly privacy-preserving) system (Sec. 6). This enables us to draw conclusions about which pathways are inherent to the functionality, and how harm prevention can be achieved through restrictions on the functionality itself.

4 Formal Model of the EUDIF

As the first building block of our methodology, and in order to rigorously model system components that could lead to harms, we provide a modular, implementation- and use-case-agnostic formal model of the EUDIF. Our model captures a class of systems that satisfy the proposed functionality of the EUDIF, as described in the eIDAS 2.0 regulation (consolidated version) [40], the implementing regulation 2025/848 [23], and the Architecture and Reference Framework (v2.8.0) [34]. We formalize the EUDIF as a functionality $\mathcal{F}_{\mathsf{EUDIF}}$ (Fig. 3), which captures privacy- and security-relevant behaviour of EUDIF-compliant systems. $\mathcal{F}_{\mathsf{EUDIF}}$ interacts with issuers $\mathcal{I}_{\mathsf{iid}}$, users $\mathcal{U}_{\mathsf{uid}}$, and relying parties (RPs) $\mathcal{R}_{\mathsf{rid}}$. It maintains a global state of issued attributes per user, disclosure policies associated with these attributes, and the registered information and intended uses for each RP. The functionality provides interfaces for RP registration and deregistration, attribute issuance, and attribute presentation.

$\mathcal{F}_{\mathsf{EUDIF}}$ captures issuers that provide users with authenticated attributes and RPs that request users' authenticated attributes. Like the EUDIF, our model allows for many issuers and RPs, and does not restrict which attributes can be issued and requested. Our model captures both previous and current versions of the EUDIF, and is expressive enough to capture all currently proposed use cases [27] and to reason formally about privacy properties thereof. Additionally, our model captures any DIF with a similar interface, such as mobile driving licenses in the United States, which are based on the same underlying standard [6, 41].

General considerations. The EUDIF shall ensure *security by design* [40, Ch. II, Sec. 1, Art. 5a 12], and its interoperability framework "shall facilitate the implementation of privacy and security by design." [40, Art. 12 3(c)]. To capture this requirement, we model a *minimal functionality* that reveals only information explicitly needed to enable the intended operation of the EUDIF and does not permit any additional leakage. This functionality captures an ideal privacy-by-design approach, i.e., the most privacy-preserving version of the system possible. We assume secure and authenticated communication channels between parties to remove leakage to third parties beyond outputs required by the functionality.

The EUDIF shall "not allow [issuers] or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user [and] enable privacy preserving techniques which ensure unlinkability, where the attestation of attributes does not require the identification of the user" [40, Art. 5a 16(a),(b)]. We model unlinkability by not notifying issuers of presentation events, and by revealing only the requested attributes.

An *attribute* attr_i is a "characteristic, quality, right or permission of a natural or legal person or of an object" [40, Art. 3 (43)]. We assume a global list of attribute descriptions, where $\mathsf{attr_descr}_i$ uniquely identifies a type of attribute (e.g., "date of birth"); this is compatible with the EUDIF interoperability requirements. The regulation foresees the authentication of attributes [40, Art. 3 (44)], which we model as *unforgeability* and *untransferability* of credentials, by managing all issued attributes through the global state Wallet.

Relying party registration. RPs should register with a national register "the intended use of European Digital Identity Wallets, including an indication of the data to be requested by the RP from users." [40, Art. 5a (2)], or more precisely, the type of services it provides, the attributes it will request, and the intended use of these attributes [23, Ann. I, Recitals 8–10]. We model this with RP and usage (de)registration routines.

Issuance. Issuers certify attributes for a user. The regulation establishes degrees of confidence in the authenticity of a user's attributes an issuer can provide (levels of assurance), without specifying how the issuer should verify the attributes it issues [40, Art. 8]. We abstract this verification process as an *issuance oracle* O_{Issue} (Step 1) to verify, e.g., the identity of a user, or the value of their attributes. When issuing attributes (Step 2), the issuer may specify a *disclosure policy*, which informs users about the conditions under which they should disclose the attribute to RPs (according to the issuer's preferences). The functionality stores the policy and attributes (Step 2). The Architecture and Reference Framework (ARF) [34] does not precisely define the scope of disclosure policies, and it is not clear if disclosure policies may receive as input any information beyond the identity of the RP and the requested attributes. The only given example of such a disclosure policy advises the user to refuse disclosure if the RP is not from a specified set of trusted RPs [34, 6.6.2.7.2]; policies of this kind must be supported as per [23, Ann. III]. We allow disclosure policies to depend on the claimed intended use and the attribute descriptions (see below, Presentation, Step 4).

Presentation. The EUDIF mandates support for *selective disclosure* of attributes, i.e., the ability for an RP to query a subset of the user's attributes [40, Art. 5a 4(a)]. We model selective disclosure by (i) considering attributes separately during the issuance and presentation phases, thereby allowing fine-grained attribute disclosure (e.g., disclose only a first name); and (ii) allowing the RP to specify a function f to be applied on the user's attributes. The function f can model simple conditions (e.g., "age above 18"), direct disclosure of attributes, or a combination thereof.

When an RP requests data from a user, the functionality and the user interact over several rounds, during which the user learns more about the request and the RP, and decides whether it approves the request. This models that users should be able to verify the registration data of RPs [23, Recital 9] and are given the possibility to check whether the RP is registered for this request [23, Recital 10, Art. 8(c)].

In Step 1, the RP sends the requested data $(f, (\mathsf{attr_descr}_i)_{i\in[n]})$ [34, 6.6.3.5.4] along with the auxiliary data aux. In Step 2, if the auxiliary data contains the claimed int_use and RP information rp_info [34, 6.6.3.5.3], then these values are stored. Otherwise, if

Parties. Issuers $\mathcal{I}_{\mathsf{iid}}$, Users $\mathcal{U}_{\mathsf{uid}}$, Relying Parties $\mathcal{R}_{\mathsf{rid}}$.
State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and attribute description to an issuer ID and issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\mathsf{iid}}$ to $\mathcal{U}_{\mathsf{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\mathsf{rid}}$;
- RPReg : rid $\mapsto$ $[(\mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\mathsf{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\mathsf{rid}}$: set RPInfo[rid] $\coloneqq$ rp_info.
RP usage registration. On input $(\mathsf{RegRP}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\mathsf{rid}}$: require RPInfo[rid] to be set; append $(\mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ to RPReg[rid].
RP usage deregistration. On input $(\mathsf{DeregRP}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\mathsf{rid}}$: require RPInfo[rid] $\neq \perp$; remove $(\mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ from RPReg[rid]. RPs can update their registration by de- and re-registering.
RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\mathsf{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] $\coloneqq \perp$.
Issuance. On input $(\mathsf{Issue}, \mathsf{iid}, (\mathsf{attr_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\mathsf{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow \mathcal{O}_{\mathsf{Issue}}(\mathsf{iid}, \mathsf{uid}, (\mathsf{attr_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\mathsf{attr})_{i\in[m]}$ and a disclosure policy policy.
(2) Set Policies[uid][iid] $\coloneqq$ policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] $\coloneqq (\mathsf{iid}, \mathsf{attr}_i)$. Send $((\mathsf{attr_descr}_i)_{i\in[m]}, \mathsf{policy})$ to the user $\mathcal{U}_{\mathsf{uid}}$ and issuer $\mathcal{I}_{\mathsf{iid}}$.

Presentation. On input (Req, rid, service) from $\mathcal{U}_{\mathsf{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Req, service) to $\mathcal{R}_{\mathsf{rid}}$, and receive $(\mathsf{aux}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\mathsf{rid}}$.
(2) If aux = (rp_info, int_use), store aux; Otherwise, if aux = $i_{\mathsf{int_use}}$, set rp_info $\coloneqq$ RPInfo[rid] and int_use $\coloneqq$ RPReg[rid][$i_{\mathsf{int_use}}$]; Otherwise, abort. Send $(\mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\mathsf{uid}}$.
(3) Receive b_{check} from $\mathcal{U}_{\mathsf{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\mathsf{check}} = 1$:
    Set $b_{\mathsf{reg}} \coloneqq 1$ if $(\mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ is in RPReg[rid], and $b_{\mathsf{reg}} \coloneqq 0$ otherwise.
    Send b_{reg} to $\mathcal{U}_{\mathsf{uid}}$, and receive $\mathsf{consent}_1$ from $\mathcal{U}_{\mathsf{uid}}$. If $\mathsf{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.
(4) For $i \in [n]$: retrieve $(\mathsf{iid}_i, \mathsf{attr}_i) \coloneqq$ Wallet[uid][attr_descr$_i$] and $\mathsf{policy}_i \coloneqq$ Policies[uid][iid$_i$].
    Send $b_{\mathsf{pol}} \coloneqq \bigwedge_{j\in[n]} \mathsf{policy}_j(\mathsf{rid}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\mathsf{uid}}$ and receive $\mathsf{consent}_2 \in \{0, 1\}$ from $\mathcal{U}_{\mathsf{uid}}$. If $\mathsf{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\mathsf{rid}}$. Otherwise, compute the output $v \coloneqq f(\mathsf{attr}_1, \ldots, \mathsf{attr}_n)$ and send v to $\mathcal{R}_{\mathsf{rid}}$.

Figure 3: Functionality $\mathcal{F}_{\mathsf{EUDIF}}$.

the auxiliary data is the index $i_{\mathsf{int_use}}$ of a registered intended use, then both rp_info and int_use are retrieved from the registry. The functionality sends all the information required for a presentation (RP information, intended use, requested data) to the user. In Step 3, the user may decide ($b_{\mathsf{check}} = 1$) to check whether the requested data have been registered by the RP [34, 6.6.3.3]. If the user decides to do so, the functionality checks (Step 3) whether the requested data $(\mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ are registered for this RP, and sends the result b_{reg} to the user. The user can then abort the presentation (using $\mathsf{consent}_1 = 0$). In Step 4, the functionality checks the RP's request against the issuer policy and informs the user of the outcome b_{pol}. The user may decide to abort the presentation with $\mathsf{consent}_2 = 0$. If the user has not instructed the functionality to abort, the functionality computes the output of the function f on the user's attributes and sends it to the RP. Our model does not specify *how* the user selects the decision bits b_{check}, $\mathsf{consent}_1$, and $\mathsf{consent}_2$.

Scope. To keep our model tractable, we exclude some aspects appearing in the EUDIF documentation that cover deployment considerations (e.g., device binding or revocation) or increase functionality (e.g., pseudonyms). These aspects are not necessary to obtain the core functionality of the EUDIF. These features can only increase leakage, and thus potential for harm, relative to what is modelled in $\mathcal{F}_{\mathsf{EUDIF}}$ (see Sec. 7).

5 Harms from EUDIF Use Cases

Following our methodology, we build upon our formal model to create *harm trees* that allow us to identify pathways to harm and their origins (see Sec. 3.2). Harms stem from elements of the system design (e.g., information leaked by the functionality, requirements imposed on users) that are used against the interests of users. However, it is challenging to reason about harms by analysing these elements of $\mathcal{F}_{\mathsf{EUDIF}}$ in isolation. For instance, without context determining *how* information is (intended to be) used and by *whom*, it is not possible to identify harms that might result from information leakage. Thus, we analyse harms arising from *specific use cases* of the EUDIF. We do not claim completeness of the harms, capabilities, or system components identified in our analysis. Yet our methodology and our conclusions remain valid: additional harms, or additional pathways to identified harms, only provide further evidence of the surface of harms from an EUDIF deployment.

Many uses cases for the EUDIF have been proposed [27] (opening bank accounts, age verification, accessing ePrescriptions, mobile driving licenses, etc.); see App. C for a discussion of use cases. All of these use cases can be modeled as a functionality in the

$\mathcal{F}_{\text{EUDIF}}$-hybrid model: the use case functionality makes calls to the operations provided by $\mathcal{F}_{\text{EUDIF}}$.

We select two use cases commonly used as flagship examples of DIFs: *age verification* and *mobile driving licenses*. These use cases also represent extremes in terms of information that must be revealed to RPs (one bit about age vs. identity). For each use case, we define a functionality based on existing official documentation and standards [27, 41] to extract system components and derive harm trees. We then identify which pathways to harm may be prevented by PETs.[1]

5.1 Harms from Age Verification

Age verification requires users to prove their age is above or below a certain threshold to a service provider using verified information issued by a trusted party such as a national government. Users may need to perform such verification to access services for which regulation requires users to be of age (e.g., adult content or gambling websites), or to access children-only spaces. This is one of the main use cases for the EUDIF [24–26]. Below, we describe how we capture the envisioned implementation of age verification via the EUDIF [26] in the functionality $\mathcal{F}_{\text{AV}}$ (see Fig. 4). Then, we extract **functionality components** from our model and build harm trees for the age verification use case.

Modeling Age Verification for Online Services. First, a central entity determines which content requires age verification, e.g., through regulation. We model this as a *content authority* CA, who mandates that an RP $\mathcal{R}_{\text{rid}}$ should register with a registration authority (Fig. 4, *Determine content requiring age verification*). RPs are not necessarily trusted. Our functionality describes RPs as separate entities from the issuer, although there may be age verification scenarios where an RP is the same entity as the issuer. We model the RP's decision on whether to register with the authority as an oracle $\mathcal{O}_{\text{reg}}$, capturing that an RP may opt not to implement age verification. The registration process itself is defined in $\mathcal{F}_{\text{EUDIF}}$.

We parametrize $\mathcal{F}_{\text{AV}}$ with a function f that captures the mandate of content authorities to verify that users are older or younger than a threshold age (modelled as an integer value n). For ease of exposition, $\mathcal{F}_{\text{AV}}$ assumes that all RPs have the same legal requirement, i.e., share the same function. A user may request issuance of a proof of age attestation from an issuer (Fig. 4, *Request issuance*), calling *Issuance* from $\mathcal{F}_{\text{EUDIF}}$. The issuer $\mathcal{I}_{\text{AV}}$ is a trusted authority, such as a government.

To access a service, the user must prove that their age is above (or below) a pre-defined threshold, i.e., provide a proof of a binary function of a single attribute. A failure in this proof results in a denial of access to the service. The RP requests an age credential from the user, captured by a subroutine call to *Presentation* in $\mathcal{F}_{\text{EUDIF}}$. We abstract the user's decision of whether to check the RP's registration status (resp., whether to engage with the age verification system based on the information they are provided from the RP) as an oracle $\mathcal{O}_{\text{check}}$, (resp., $\mathcal{O}_{\text{eng}}$). If the user consents, the RP is provided with the output of the verification function f.

[1]We consider PETs that can be implemented within the EUDIF (e.g., anonymous credentials, data federation), and that do not depend on infrastructural changes. PETs outside the implementation of the EUDIF (e.g., accessing services via Tor, alternative Internet architectures) are out of scope.

The oracle $\mathcal{O}_{\text{ver}}$ captures the RP's decision to grant access to the user given the output of f. We do this rather than hard-coding granting access when age verification succeeds, as in practice, it is impossible to force RPs to provide access if verification passes. We do not model fallback mechanisms for when the initial verification does not succeed, e.g. asking for *more* information such as the user's date of birth, age in years, or to show their complete identity information as suggested in the ISO standard [41]. Implementing such fallbacks would only increase the information leakage of the system, and therefore potentially result in more harms.

Harm Trees for Age Verification. We construct harm trees for the harms that arise from the use of the EUDIF for online age verification. We extract potential harms from existing literature [37, 49, 50] and expand this set with harms discussed in [12, 15]. The harms we consider are relative to a baseline scenario where age verification is not widely deployed, as is current practice in the EU. We find seven harms: discrimination, exacerbation of structural exclusion, targeted advertising harms, reduced availability of resources, censorship, diminished Internet safety, and increased cybersecurity risk. We discuss the first three here and defer the discussion of the rest to App. E. For system components that are inherent to the minimal functionality of the use case (**functionality components** in **black**), we specify the corresponding element of the ideal functionality $\mathcal{F}_{\text{AV}}$. Recall that we also extract relevant **digital-context components** and **implementation components** (corresponding to the naive implementation where the user reveals all attributes upon each presentation).

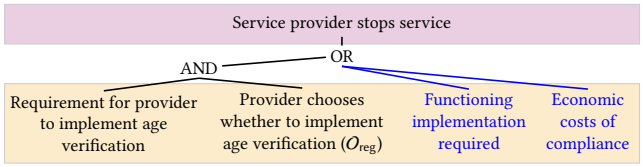
Common sub-trees. We surface two capabilities that are common to several harms: ceasing service and tracking (Fig. 5). For clarity, we explain these sub-trees first.

Service provider stops service (Fig. 5a). Due to implementation challenges or fear of legal consequences, service providers might decide to not implement age verification and either *stop* providing a service altogether or pull out of specific regions. Legal requirements to implement age verification are inherent to the core functionality: a service provider receives a mandate from the Content Authority and must register to provide a service (*Requirement for provider to implement age verification,* in Fig. 4, i.e., **functionality components**). Implementing age verification in a digital context might necessitate expertise to develop a functional implementation, which might be unavailable to some service provider or incur high costs (i.e., **digital-context components**).

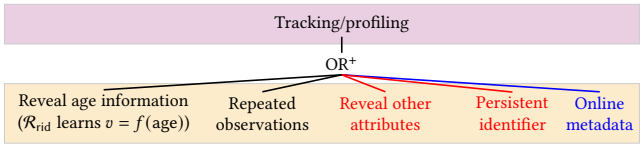
Tracking and Profiling (Fig. 5b). Information leakage to the RP can enable service providers to track and/or profile users. In this use case, the only information leakage to the RP that is inherent to the use case functionality is the result of the function $f(\text{age})$. The RP might learn additional information (e.g., online metadata or personal information) from the digital context of deployment (e.g., a browser or application). This online metadata cannot be eliminated with modifications to the implementation of the EUDIF itself (e.g., use of PETs), because such modifications do not affect the digital context in which it is used (e.g., over the Internet). In the naive, concrete implementation where users' identity is revealed, the presence of a persistent identifier trivially enables tracking and profiling. Many combinations of these components suffice to grant

Parties. Issuer $\mathcal{I}_{\mathsf{AV}}$ is a trusted authority (e.g., a government) issuing age credentials to users (we set iid := AV). Users $\mathcal{U}_{\mathsf{uid}}$ are individuals accessing online services. Relying Parties $\mathcal{R}_{\mathsf{rid}}$ are online service providers/operators. A content authority CA (e.g., European Commission, government) determines which content requires verification.
Parameters. $f_{>n}$ is a function that computes age $> n$ (resp. $f_{<n}$ for age $< n$), e.g., $f_{>16}$, $f_{<18}$, or $f_{>21}$. $O_{\mathsf{reg}}(\mathsf{rid})$ is an oracle that determines whether $\mathcal{R}_{\mathsf{rid}}$ registers as an RP. $O_{\mathsf{check}}(\mathsf{rid}, \mathsf{intended_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, \mathsf{uid})$ is an oracle that determines whether $\mathcal{U}_{\mathsf{uid}}$ checks $\mathcal{R}_{\mathsf{rid}}$'s registered information. O_{eng} is an oracle that on input $(b_{\mathsf{reg}}, \mathsf{uid})$ or $(b_{\mathsf{pol}}, \mathsf{uid})$ outputs a (consent) bit $\mathsf{consent}_1$ or $\mathsf{consent}_2$ determining whether $\mathcal{U}_{\mathsf{uid}}$ consents to engage with the system. $O_{\mathsf{ver}}(v, \mathsf{rid})$ is an oracle that determines whether $\mathcal{R}_{\mathsf{rid}}$ grants access to the service based on the presentation output v.
Determine content requiring age verification. On input (RequireAge, rid) from CA, send Reg from $\mathcal{R}_{\mathsf{rid}}$.
RP registration. On input Reg from $\mathcal{R}_{\mathsf{rid}}$, get $b \leftarrow O_{\mathsf{reg}}(\mathsf{rid})$. If $b = 1$, send (RegRP, age-verif, f, age) from $\mathcal{R}_{\mathsf{rid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$.
Request issuance. On input (RequestIssuance, uid) from a user $\mathcal{U}_{\mathsf{uid}}$: Send (Issue, av, age) from $\mathcal{U}_{\mathsf{uid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$.
Access service. On input (AccessService, rid) from $\mathcal{U}_{\mathsf{uid}}$:
(1) Send (Req, rid, OnlineAccess) from $\mathcal{U}_{\mathsf{uid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$. Upon receipt of (Req, OnlineAccess), $\mathcal{R}_{\mathsf{rid}}$ returns $(f, \mathsf{age}, 1)$.
(2) Send (rid, age-verif, f, age, uid) to O_{check} (upon receipt from $\mathcal{F}_{\mathsf{EUDIF}}$) and return the result as b_{check} to $\mathcal{F}_{\mathsf{EUDIF}}$.
(3) Upon receipt of b_{reg} or b_{pol}, $\mathcal{U}_{\mathsf{uid}}$ forwards the input to O_{eng} and returns the result as $\mathsf{consent}_1$ or $\mathsf{consent}_2$ to $\mathcal{F}_{\mathsf{EUDIF}}$.
(4) Upon receipt of $v := f(\mathsf{age})$, $\mathcal{R}_{\mathsf{rid}}$ gets a bit $b \leftarrow O_{\mathsf{ver}}(v, \mathsf{rid})$. Return GrantAccess to $\mathcal{U}_{\mathsf{uid}}$ if $b = 1$; else DenyAccess.

Figure 4: Functionality $\mathcal{F}_{\mathsf{AV}}[\mathcal{F}_{\mathsf{EUDIF}}]$ for age verification for an online service



(a) Service provider stops service or pulls out from territory



(b) Tracking/profiling

Figure 5: Age verification – Common sub-trees of capabilities

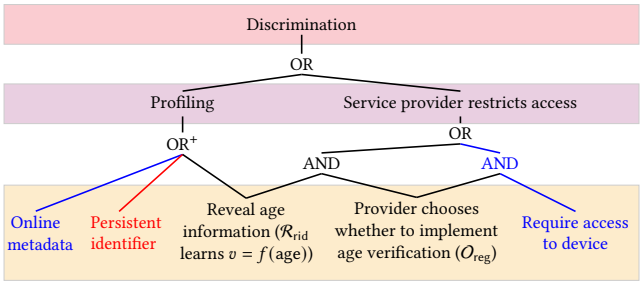
the RP tracking and profiling capabilities. In fact, even if PETs are used to minimize the information revealed and eliminate persistent identifiers, the functionality inherently reveals age information that can be used for profiling (see Sec. 6.2).

Discrimination (Fig. 6a). Discrimination harms involve entrenching inequality and disadvantaging people based on their attributes (e.g., gender, race, nationality, age) [12]. The possibility of discrimination based on age information is inherent to the functionality. While in some cases acting on age information might be legitimate (e.g., managing access to adult-only content), this information can also be used to provide discriminatory service (e.g., discriminatory pricing). Combined with online metadata due to the EUDIF's digital context (e.g., cookies, fingerprinting, advertising IDs), or persistent identifiers created in a naive implementation, age verification improves profiling and increases discrimination harms, e.g., due to targeted advertising [4]. Finally, the digital nature of EUDIF requires that users have access to a device and that this device has sufficient power/battery when the user is required to present a credential. This device must have adequate resources to run the concrete EUDIF implementation, which may exacerbate the risk of discrimination (see Sec. 7).

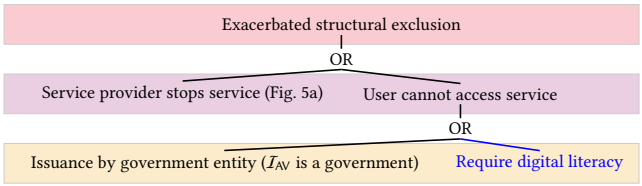
Profiling due to leakage of persistent identifiers can be mitigated by privacy mechanisms that provide unlinkability. However, PETs cannot prevent profiling based on age information and information from the digital context. We show in Sec. 6.1 that this form of discrimination is inherent to the EUDIF functionality, regardless of the use case.

Exacerbate structural exclusion (Fig. 6b). Structural exclusion restricts individuals' choices and freedom of information, and can be considered an *autonomy harm* [12]. With age verification, exclusion can happen due to two capabilities. First, users may not be able to access the service. This affects individuals who do not fulfil age requirements, but also those who cannot verify their age for other reasons (e.g., those with low digital literacy, or with no access to identity documents). Second, if service providers pull out of territories, all users from those territories lose access to the service, e.g., [16, 48, 62].

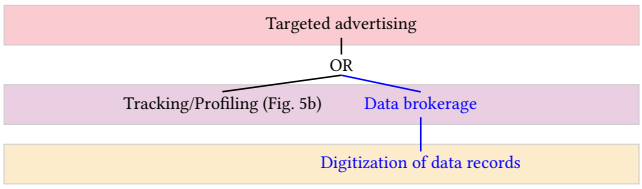
Targeted advertising harms (Fig. 6c). Targeted advertising has been shown to lead to psychological harm [44, 52], result in discriminatory practices [4], and restrict users' opportunities for choice and control over their actions [59, 68]. Using the EUDIF for online age verification is set to further deepen these harms. Minors can be targeted for advertisements based on verified age information that must be revealed to achieve the desired functionality. Adults can be targeted based on their verified age and information leaked through the digital context (directly or via data brokerage). A naive implementation increases the risks of targeting by enabling easier tracking and profiling (see Fig. 5b). Even if PETs are used to limit profiling, the risk of targeted advertising harms persists due to the age-related leakage.



(a) Discrimination

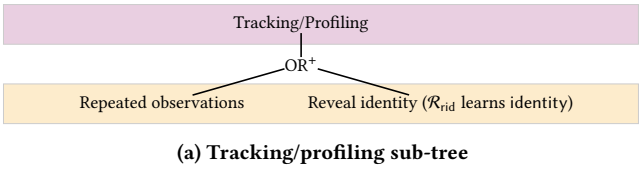


(b) Structural exclusion

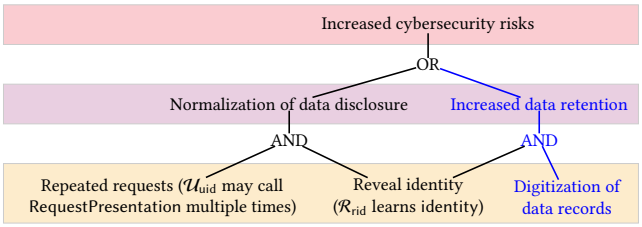


(c) Targeted advertising

Figure 6: Age verification harm trees



(a) Tracking/profiling sub-tree



(b) Increased cybersecurity risks

Figure 7: Mobile driving license (mDL) harm trees

5.2 Harms from Mobile Driving Licenses

A *mobile driving license* (mDL) is a digital document that proves an individual's right to drive. In certain jurisdictions an mDL can include information such as a full name, photo, date of birth, and nationality, and can also be used as a means of digital identification. mDLs are proposed as a use case for the EUDIF for: (i) verification of a driver's right to drive when renting cars, (ii) identification upon police roadside checks, and (iii) as a means of electronic identification more generally. We consider the first use case: a customer visits a car rental service (in person) to pick up a car, and a clerk requests their identification and driving license. The user reviews the presentation request, and if they consent to engage with the system, confirms disclosure of their identity and mDL. The clerk's verifier device validates the proof of driving rights and displays the photo corresponding to the user's credential. The clerk performs out-of-band verification of the user's identity. We formalize the use case according to the mDL ISO standard [41]. We defer the details to App. D.

Harm Trees for Mobile Driving License. We extract potential harms from the literature [66] and expand them with our own analysis. We then create harm trees based on the prior mDL use case description. The baseline usual context against which we compare is a user presenting their physical identity document. We find that several of the harms discussed for age verification apply verbatim to this use case, and some pathways to harm are simplified. For all trees that change with respect to the age verification case, there are no pathways depending on concrete design decisions (**implementation components**). We describe the most notable differences here.

First, the discrimination harm tree (Fig. 6a) applies verbatim to the mDL scenario: the car rental service can choose to discriminate based on the presented identity, which may include attributes such as age or nationality. Despite being primarily an *offline* interaction, a digitalized mDL presents the same harm tree for exacerbated structural exclusion (Fig. 6b). The mDL use case still requires the user to have access to a device, digital literacy skills, and official identity documents.

Second, *targeting* and *profiling* of users is easier in the mDL case than with age verification. Attributes that, for age verification, could be protected using PETs (i.e., **implementation components**) cannot be hidden in the mDL use case because the functionality requires that users reveal their identity to the provider. Thus, an RP observing repeated presentations (e.g., when an individual rents a car again or uses their mDL for another use case) can track their behaviour more easily. Fig. 7a illustrates the simplified pathways to the tracking and profiling capability. This sub-tree of tracking and profiling applies to all identity-based use cases in Table 1.

Lastly, digitization of identity documents may lead to greater retention of data which increases the risk of sensitive data being shared or leaked in data breaches or voluntarily by the data holder. Regular use of digital identities and mDLs may normalize requests for verified identity information online and in proximity cases. This leads to broader sharing of personal information that heightens users' vulnerability to phishing, identity fraud, and data breaches. Thus, mDLs increase cybersecurity risks which might lead to downstream harms such as economic and psychological harms (Fig. 7b). Each of these identified harms are either exacerbated by or introduced by the use of the EUDIF, compared to the baseline scenario where one presents a physical identity document.

6 Fundamental Harm-enabling Capabilities

We now use our adversarial capability games (see Sec. 3.1) to identify *fundamental harm-enabling capabilities*, the second goal of our

harm analysis (see Sec. 3.2). We provide a deeper analysis of three capabilities that lead to most of the harms we identified via our harm trees: *distinguishing*, *tracking*, and *inference*. Using the games, we define which system components lead to these capabilities, identify attack strategies to exploit them, and gain an understanding of the root causes of these capabilities.

6.1 Distinguishing Capability

Informally, distinguishability is the ability of an RP to distinguish between *classes of users*, and thus categorize them based on a single presentation of their wallet. We formally model an adversary's capability to distinguish users using a security game against the $\mathcal{F}_{\mathsf{EUDIF}}$ functionality (App. F.1, Fig. 15). The adversary's goal is to distinguish between two fresh challenge users. Freshness enables us to model the adversary's advantage in distinguishing between users *it has not seen before*. Our tracking game in the next section captures the ability to track (or link) a user between different events.

The game proceeds in two phases. First, the adversary specifies which attributes and values should be issued and the disclosure policies for each of the two fresh users. We model issuance as honest but let the adversary determine the policies for the fresh users. These adversarial policies capture leakage that occurs when issuers use different policies for different users, e.g., stricter policies for underage users. Second, the adversary, acting as a malicious RP, can request a presentation from one of the two fresh challenge users. The adversary can choose the function f to determine which (function of the) attributes the user must disclose. To model the diversity of users' decisions, the adversary can specify the behaviour of both challenge users with respect to whether they require registration of the RP and whether they consent.

This distinguishing game gives the adversary freedom to model users that are, somehow, different: the adversary can control for the challenge users (a) the policies, (b) the attributes and/or disclosed values, or (c) the behaviour with respect to registration checks and consent. The distinguishing game has three variants (policies, attributes, behaviour) where the adversary is permitted to induce differences in only one of these three controls. These three variants might not be the only ways that an adversarial RP can distinguish users (e.g., adversaries might combine controls). Yet, they let us reason about how different features of the $\mathcal{F}_{\mathsf{EUDIF}}$ functionality contribute to the ability of the RP to distinguish users.

Results. Our formal analysis in App. F.1 shows that adversaries can exploit each of these three controls through $\mathcal{F}_{\mathsf{EUDIF}}$ to distinguish between users, and thus, that the adversary's ability to distinguish users is inherent to $\mathcal{F}_{\mathsf{EUDIF}}$ itself.

First, adversaries can exploit that different users have been issued different policies. The outcome of this policy feeds into the user's consent decision-making process. For example, if children are given a policy not to disclose their first name, an RP that asks for a user's first name can distinguish children.

Second, adversaries can exploit that different users have different attribute values. This captures the case where the evaluations of the function f on the attributes differ between the two challenge users. For example, an RP might ask for a users' nationality, and thus distinguish users based on this.

Third, adversaries can exploit differing user behaviour with respect to consent and registration checking. By asking for invasive attributes, the adversary can cause some users to not consent, leading to a mechanism to distinguish users. For example, non-binary users might be less willing to reveal their gender [56, 61].

Takeaway. While there could be legitimate and fair reasons to enable distinguishing capabilities, the existence of this capability is inherently a source of harms such as discrimination (Fig. 6a), exacerbated structural exclusion (Fig. 6b), reduced availability of resources (Fig. 11), and censorship (Fig. 12). Our analysis demonstrates that even an ideal EUDIF implementation would lead to this capability, and that the causes are system-level decisions by participating parties.

Any deployment aiming to thwart distinguishing harms must impose restrictions on the system-level specification. In particular, limiting distinguishing requires controlling the policies that issuers apply, and which (functions of) attributes RPs are allowed to ask for. To avoid policies becoming a mechanism to distinguish between groups A and B, the policies of users in group A must evaluate presentation requests from an RP in the same way as the policies of users of group B. The easiest way to accomplish this is to forbid the use of per-user policies. To reduce the impact of information revealed by attributes and consent, a necessary condition is to limit which functions f and attributes RPs are allowed to ask for: (1) the function f must not have different outputs on two groups of users; and (2) the attributes and function f must not trigger different consent evaluations between groups of users.

In traditional cryptographic games, much of the freedom that we provide to the adversary in the distinguishing game is ruled illegal (or as trivial wins), precisely because the differences it induces cannot be prevented. We deliberately include them here because these differences between users *can* occur in a deployment of the EUDIF. These abilities must not be discarded in broader systems analyses because real adversarial RPs can indeed ask for (functions on) such kind of attributes.

6.2 Tracking Capability

Informally, tracking is the ability of the issuer and an RP to recognize a user they have seen before. Tracking may lead to harms such as discrimination (see Figs. 5b and 7a). Tracking is closely related to, but distinct from, distinguishing. To formally define tracking, we take inspiration from the cryptographic notion of *unlinkability* in anonymous credentials. In this analogue to anonymous credentials, distinguishing is to tracking as zero-knowledge is to unlinkability: the former captures the ability to learn information about the user, while the latter captures the ability to link interactions of the user.

To model tracking, we extend the distinguishing game (i) with additional adversarial queries over time (*multi-show tracking*), where an adversary seeing two presentations for the same value v guesses if they come from the same user, or (ii) by allowing the adversary to also corrupt issuers (*issuer tracking*), where an adversary seeing a presentation tries to learn the issuing event for the disclosed attributes. Our tracking games are in App. F.2 (Figs. 16 and 17).

Results. Our analysis shows that in the context of the EUDIF, an adversary achieving the distinguishing capability (through either policies, attributes, or user behaviour) also achieves the (issuer or

multi-show) tracking capability (for the same game flavour). There might exist adversaries that achieve tracking without distinguishing. In App. F.2, we show that whenever an RP is allowed to ask for a *sufficiently identifying set of attributes* it can track users across presentations (multi-show tracking), or collude with issuers to link that user's presentations to their issuing event (issuer tracking).

Takeaway. Similarly to distinguishing, tracking also stems from system-level decisions. Preventing tracking requires adding restrictions on policies, attributes, and user behaviour (Sec. 6.1). These restrictions must hold even when multiple RPs and issuer collude and share knowledge about users. Further, ensuring that the set of attributes requested is not a sufficiently identifying set of attributes is similar to enforcing k-anonymity and ℓ-diversity (for large values of k and ℓ) over all possible user attributes, which might be impossible in practice.

6.3 Inference Capability

Inference is the adversary's capability to infer user attributes. This includes attributes that the user willingly discloses to a corrupted RP, as well as other attributes that are correlated to the disclosed attributes. In our inference game (Fig. 18 in App. F.3), the adversary chooses a probability distribution of attribute tuples. This probability must be sufficiently unpredictable, i.e., have min-entropy at least κ for some parameter κ. The challenger samples an attribute tuple from this distribution, and issues corresponding credentials to a user. The adversary, corrupting a set of RPs, can (adaptively) request up to q presentations of the user's attributes. The adversary wins if it can infer information about the user's attributes. To precisely articulate what inference means, we borrow ideas from the quantified information flow (QIF) literature [5], and define three flavours for the winning condition:[2] (i) full recovery, requiring the full tuple of attributes, (ii) recovery of sensitive attributes, where sensitivity is hard-coded into the game, and (iii) g-leakage, which specifies an arbitrary predicate (e.g., partial advantage for approximate guesses, different weights for different attributes). Full recovery implies recovery of sensitive attributes, and both are a special case of g-leakage.

Results. An adversary controlling RPs can trivially recover all of a user's attributes with a single presentation query which requests all attributes in full; this attack also immediately reveals all sensitive attributes and any g-leakage function. One could consider a restriction of the EUDIF where an RP is not allowed to query all of a user's attributes but only a limited subset, and further, that the queries an RP makes must be fully specified at setup. In App. F.3, we show that even in this restricted setting, there exists an adversary that infers sensitive attributes and g-leakage, by asking for presentations through multiple RPs and requesting presentations adaptively.

Takeaway. The current vision of the EUDIF and many of its advertised benefits include a flexible "any-RP-may-ask-for-anything" policy. We show that this entails unbounded inference capabilities. Conversely, any measure that restricts the inference capability would heavily restrict the functionality of the EUDIF, and thus its anticipated benefits.

[2]These are not exhaustive, and the QIF literature contains more definitions of leakage, which in our setting capture different pathways to harms.

To meaningfully bound the maximum privacy leakage of users' information, it is imperative to restrict the adversary's ability to request arbitrary user attributes. A radical solution to prevent inference would be to disallow the issuance of credentials. This obviously negates the utility of the EUDIF, but a more nuanced credential issuance restriction may reduce harm (e.g., disallowing the issuance of credentials for medical records). Enforcing that presentation requests be limited to those registered would also help reduce harms, as long as the set of registered queries is restricted such that any combination of queries may only lead to a bounded inference advantage (e.g., forbid revealing "sensitive" attributes in conjunction with pseudo-identifiers). Yet, this restriction faces operational challenges: it requires RPs to have up to date registrations and continuous audits to check that query sets are indeed harmless. Moreover, such a measure would not help against an adversary corrupting multiple RPs, such as advertising entities embedded in many web pages, or third parties that run verification of credentials for many RPs. Such "intermediaries" are explicitly envisioned in the EUDIF [34, Sec. 3.11.3].

In summary, to restrict adversaries from achieving the inference capability, the EUDIF needs to enforce issuer disclosure policies, limit presentations to registered attributes, limit which attributes may be issued and queried, and enforce non-sharing of information between RPs. These constraints must be enforced with regulations and oversight, reduce scalability and interoperability of the EUDIF, and cannot completely prevent inference. We stress that inference is inherent to the EUDIF functionality, and can not be mitigated by PETs.

7 Harm-enabling Components from Deployment Requirements

In the above harm analysis, we use the **implementation components** stemming from a naive implementation with minimal functionality and where users send their complete attributes. In real-world deployments, additional constraints and requirements further increase the complexity of the system and its implementation. Concrete implementations may also use privacy protections to minimize the shared data. These can introduce new parties and pathways to harm beyond those identified in our preliminary analysis.

To illustrate that our methodology serves to analyse complex systems, beyond the best privacy-preserving implementation (our formal model in Sec. 4) and the worst case scenario of a naive implementation, we extract realistic deployment details and additional features from public discussions of the framework [33], the reference implementation of the EUDI wallet [36], and reports from pilot projects [35] and discuss how these changes lead to new pathways to harm.

We illustrate new flows and parties that appear from these requirements in Fig. 8. First, a device hosts the wallet to bind it to the user. Three new third parties appear: registries that hold registered RPs and their policies, a disclosure policy manager to assist users in their decision about the policy of an RP, and a revocation list manager that assists RPs in identifying invalid credentials. New flows also materialize: steps 0, 4, and 6. We illustrate the impact of these deployment requirements on harm trees using the example

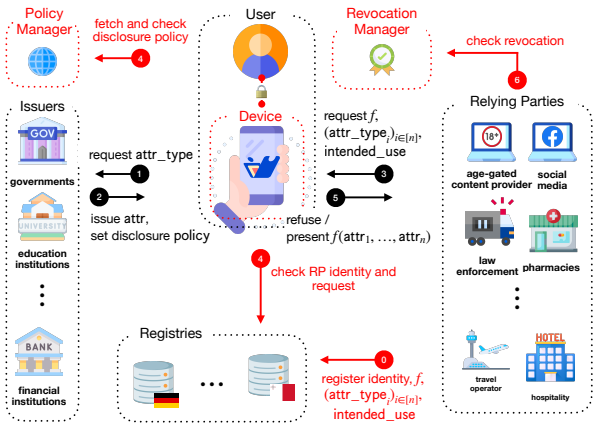


Figure 8: Additional information flows stemming from deployment requirements (in red).

of the discrimination harm tree (Fig. 6a) with its modified version shown in Fig. 19. We note that our set of deployment requirements is not complete, and other requirements may result in additional pathways to harm.

Device requirements increase discrimination risk. The digital nature of the EUDIF requires a physical device to store and operate the wallet. As shown in Fig. 6a, requiring access to a device, by itself, can lead to discrimination. When implementing the EUDIF, functionalities or implementation choices may introduce requirements on the device, such as a specific operating system (e.g., requiring particular attestation methods [45]) or the presence of a secure element (e.g., to enable device binding [42]). The need for specific device characteristics adds a new **implementation component** to the *discrimination* harm tree and exacerbates this harm.

Device binding increases exclusion risks. In many applications, such as age verification, security hinges on users not being able to share credentials. The ideal functionality does not cover this aspect. To prevent sharing of credentials, the EUDIF relies on *device binding* to link a wallet to a device [34, Sec. 6.6.3.8]. Having access to the device serves as a proof of ownership of the wallet stored on it. Wallets cannot be transferred to other devices without changing the binding to that device.This one-person-one-device ownership model does not correspond to many realistic scenarios in which users have multiple devices (e.g., smartphones and laptops, or corporate and personal devices) or share devices with other users (e.g., family computers/smartphones) [65]. In these scenarios, device binding might prevent users from using their wallet in certain situations. For example, binding the wallet to a laptop impedes its use in offline interactions such as age verification to buy certain products; binding the wallet to a (shared) phone impedes its use in online interactions on a laptop if the user does not have the phone at that point. These can exacerbate *structural exclusion* when the users' choices depend on their demographics (e.g., elderly users that are not comfortable with mobile technology) or socioeconomic status.

Additionally, the single-device policy means that users need to use their wallet-bound device when accessing services on another device. Doing so implies that they connect sessions from both devices (e.g., by scanning a QR code) which creates new **implementation** pathways for *profiling* and *tracking*.

Relying parties' certificates. To ensure that users can trust RPs, the ARF [34] introduces trusted *registrars* that can attest to the validity of registered RPs' certificates. RPs can embed these certificates in presentation requests, and the user can contact the *registrar* to check its validity. This check (step 4 in Fig. 8) may leak information about the user's interactions with the RP to the registrar, creating a new **implementation component** (*Registrar learns* $\mathcal{R}_{\text{rid}}$) in harm trees, and a new pathway to *profiling* and *tracking*.

Disclosure Policy. The ARF specifies that users must be made aware of the RP's privacy policy before providing informed consent to share credentials [34, Ann. 2.02 , EDP_05]. This policy is provided as part of the presentation request or encoded as a URL hosted by a party other than the RP, e.g., the registrar or the issuer. If the user chooses to visit the URL, in step 4 it will reveal to the party storing the policies the accessed service (*even* if later the user decides not to consent). This creates a new **implementation component** (*Policy manager learns* $\mathcal{R}_{\text{rid}}$) that contributes to *tracking*.

Revocation. Credentials can refer to facts that change over time (e.g., address, age), have an expiration time, or be compromised. Thus, issuers must have the ability to revoke credentials, and RPs must be able to verify that credentials are not revoked [34, Ann. 2.02, Topic 7]. Revocation would also be necessary to support device binding and allow transferring a wallet between devices. For instance, a user changing device could revoke their credential on the original device and re-register it on the new one. Implementing revocation requires a Revocation Manager to hold revocation information (e.g., lists of revoked credentials). The ARF requires that credentials are validated upon presentation. Depending on the implementation, this check might introduce an interaction between the RP and the Revocation Manager (step 6 in Fig. 8). This would introduce a new **implementation component** (*Revocation Manager learns the* $\mathcal{R}_{\text{rid}}$) contributing to *tracking*/*profiling*.

8 The Path towards Harm Prevention

To conclude, building on our findings in previous sections, we discuss potential solutions to prevent harms.

8.1 Effectiveness of PETS in Limiting Harms

The EUDIF documentation emphasizes a privacy-by-design approach [34], partly to address feedback concerning its potential negative impact on users [9, 21]. Based on our findings, we assess the potential of PETs to succeed at this goal.

PETs can eliminate a subset of pathways to harms. PETs hinder adversarial capabilities, making certain harms that stem from digital identity systems more difficult to bring about. Implementing PETs is therefore highly recommended; however, the extent to which PETS can reduce harm-enabling capabilities is limited. Trivially, integrating PETs in an implementation of the EUDIF *cannot eliminate* the presence of components in the core **functionality**. By necessity,

these components can be as leaky as revealing the full identity of the user (e.g., in the mDL). Neither can integrating PETs eliminate harms that arise from the **digital context**, such as those arising from the need for a device to access services or the digital literacy required to use a wallet. At most, PETs can remove components in EUDIF implementations arising from **implementation** decisions (e.g., anonymous credentials with selective disclosure eliminate the *Reveal other attributes* component in the tree in Fig. 5b). Removing these components brings the leakage of the implementation closer to that of the (ideal) EUDIF functionality in a digital context. However, in each harm tree, there exists at least one pathway to the harm that does not require an **implementation component**; PETs thus cannot, by themselves, eliminate the potential of the EUDIF and other digital identity systems to induce harms.

Using PETs does not guarantee minimization of adversarial capabilities. For example, batch issuance of credentials, as proposed in the Swiss electronic ID [8], does not guarantee unlinkability. Even though it does not create a global 'persistent identifier' (see Fig. 5b), it enables issuers (colluding with an RP) to track users. In this case, the pathway to harm remains, although the tracking capabilities may be more challenging to attain than in the naive implementation. Similarly, pseudonyms enable users to access service in a linkable manner without revealing their identity [51]. However, given that linkability within (and potentially across) visits to an RP persists with the pseudonym, the *profiling* and *discrimination* capabilities also persist.

PETs can introduce or exacerbate pathways to harm. Modern PETs are complex and can be opaque to users [10, 14, 46, 47, 63, 67]. This opacity may lead to new system components in harm trees stemming from users' actions. For example, users might refuse to use a system that they do not understand, adding a new **implementation component** to the *Structural exclusion* harm tree (Fig. 6b); or users might misunderstand the protection and reveal more information than intended, resulting in the component *Reveal other attributes* persisting in the *Tracking/Profiling* sub-tree (Fig. 5b) or the component appearing in the *Discrimination* tree (Fig. 6a).

PETs also change the requirements on devices: they increase computational power requirements on the user's device (e.g., zero-knowledge proofs underlying zkSNARKs may be too resource-intensive to run efficiently on low-end or older smartphones [43, 57]), or may require a trusted execution environment (which not all devices have), which amplifies the harm of *Structural exclusion* (Fig. 6b). PETs might need specialized optimizations to meet the limitations of users' devices (e.g., storage, hardware) and usability constraints. Only a few actors have the expertise and means to design, implement, and maintain such specialized PETs (e.g., Big Tech players). Such concentration of power gives these actors control of, e.g., the libraries' output formats or APIs, reducing freedom of choice (e.g., the system has to be compatible with the imposed implementation), undermining *autonomy*.

Identity use cases have inherent harms. Harms can arise from the deployment of identity-based use cases, such as mDLs and many others in App. C, regardless of the use of PETs. Our analysis identified that, for the mDL harm trees (and thus, for related identity use cases), no pathways to harm relied on **implementation components**. The need to share identity eliminates the ability to achieve unlinkability. Use of the EUDIF will exacerbate the capabilities of tracking and profiling, relative to a scenario where a physical ID is used.

8.2 Policy Restrictions can Eliminate Harms

Our analysis identifies three harm-enabling capabilities that are inherent to the EUDIF functionality: distinguishing, tracking, and inference. Even with PETS, a wide range of harms remains possible.

The analysis also reveals that restricting the system functionality can mitigate their negative impact. Thus, the *only* way to reliably reduce the harm surface in digital identities is to constrain the system via regulatory and procedural controls. In the case of the EUDIF, rules and processes that govern how it is used can eliminate the harms that we have identified. Restrictions on disclosure policies, attributes, and user and RP behaviours can thwart distinguishing and tracking capabilities. This requires exerting tight control over issuers and RPs to ensure the system-level restrictions are followed. And to limit the inference capability it is necessary to limit the utility of the EUDIF itself.

We began this work by asking whether PETs are an appropriate answer to concerns about the negative societal impacts of the EUDIF. Given that the impact of PETs on harm-enabling capabilities is limited, we conclude that promoting PETs as the main solution to mitigate negative societal impacts might result in a false sense of safety and prevent a broader discussion about the desirability of digital identity systems and their uses. Instead, the path towards a safer and more inclusive digital space is to complement the use of PETs with constraints on the scope of digital identities, potentially reducing their benefits.

We hope that our findings help broaden the conversation between technologists, policymakers and society, from discussing improvements of the privacy of current proposals to the benefits and costs of scope-reduced digital identities.

References

[1] 119th United States Congress. S.1748 - Kids Online Safety Act . https://www.congress.gov/bill/119th-congress/senate-bill/1748, 2025. URL: https://www.congress.gov/bill/119th-congress/senate-bill/1748.

[2] 119th United States Congress. H.R.8250 - Parents Decide Act . https://www.congress.gov/bill/119th-congress/house-bill/8250, 2026. URL: https://www.congress.gov/bill/119th-congress/house-bill/8250.

[3] Transportation Security Administration. Participating states and eligible digital IDs, 2025. https://www.tsa.gov/digital-id/participating-states.

[4] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. Discrimination through optimization: How Facebook's ad delivery can lead to biased outcomes. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), November 2019. doi:10.1145/3359301.

[5] Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In Steve Zdancewic and Véronique Cortier, editors, *CSF 2012 Computer Security Foundations Symposium*, pages 265–279. IEEE Computer Society Press, 2012. doi:10.1109/CSF.2012.26.

[6] American Association of Motor Vehicle Administrators (AAMV). Mobile driver's license (mdl) implementation guidelines, 2025. https://www.aamva.org/getmedia/bb4fee66-592d-4d39-813a-8fdfd910268a/MobileDLGuidelines1-5.pdf.

[7] Myrto Arapinis, Véronique Cortier, Hubert de Groote, Charlie Jacomme, and Steve Kremer. Are ideal functionalities really ideal? Cryptology ePrint Archive, Report 2025/2125, 2025. URL: https://eprint.iacr.org/2025/2125.

[8] Swiss Federal Authorities. Swiss' electronic identity (e-id), 2025. URL: https://www.eid.admin.ch/en/e-id-e.

[9] Carsten Baum, Olivier Blazy, Jan Camenisch, Jaap-Henk Hoepman, Eysa Lee, Anja Lehmann, Anna Lysyanskaya, René Mayrhofer, Hart Montgomery,

Ngoc Khanh Nguyen, Bart Preneel, abhi shelat, Daniel Slamanig, Stefano Tessaro, Søren Eller Thomsen, and Carmela Troncoso. Cryptographers' feedback on the EU digital identity's ARF. Technical report, 2024.

[10] Zinaida Benenson, Anna Girard, Ioannis Krontiris, Vassia Liagkou, Kai Rannenberg, and Yannis C. Stamatiou. User acceptance of privacy-abcs: An exploratory study. In *Human Aspects of Information Security, Privacy, and Trust*, volume 8533 of *Lecture Notes in Computer Science*, pages 375–386, Cham, 2014. Springer.

[11] Ran Canetti, Julie Ha, and Gabriel Kaptchuk. Analyzing cryptography in context: A cryptography-native approach to threat modeling. In Elissa Redmiles and Ben Stock, editors, *USENIX Security 2026: 35nd USENIX Security Symposium*, pages 1987–2006, Baltimore, MD, USA, August 12–14, 2026. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity26/presentation/canetti.

[12] Danielle Keats Citron and Daniel J Solove. Privacy harms. *Boston University Law Review*, 102:793, 2022.

[13] EU Commission. European Digital Identity — commission.europa.eu. https://commission.europa.eu/topics/digital-economy-and-society/european-digital-identity_en, 2026. [Accessed 24-04-2026] Archived at https://web.archive.org/web/20260419112417/https://commission.europa.eu/topics/digital-economy-and-society/european-digital-identity_en.

[14] Rachel Crowder, George Price, and Thomas Groß. Simply tell me how – on trustworthiness and technology acceptance of attribute-based credentials, 2023. URL: https://arxiv.org/abs/2308.06555, arXiv:2308.06555.

[15] CyberPeace Institute. Report of second expert meeting on harms methodology, 2024. URL: https://cyberpeaceinstitute.org/wp-content/uploads/2024/10/Second-Expert-Meeting-Harms-Methodology-2024.docx.pdf.

[16] Liam Dawe. The gamingonlinux forum is shutting. *GamingOnLinux*, December 2024.

[17] Sourya Joyee De and Daniel Le Métayer. PRIAM: A privacy risk analysis methodology. In Giovanni Livraga, Vicenç Torra, Alessandro Aldini, Fabio Martinelli, and Neeraj Suri, editors, *Data Privacy Management and Security Assurance*, pages 221–229, Cham, 2016. Springer International Publishing.

[18] Sourya Joyee De and Daniel Le Métayer. Privacy harm analysis: A case study on smart grids. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 58–65. IEEE, 2016.

[19] Sourya Joyee De and Daniel Le Métayer. A refinement approach for the reuse of privacy risk analysis results. In *Annual Privacy Forum, APF*, pages 52–83. Springer, 2017.

[20] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.

[21] epicenter.works. Analysis of 5th eidas compromise from czech council presidency, October 2022. https://epicenter.works/fileadmin/import/analysis_of_5th_eidas_compromise_from_czech_council_presidency_0.pdf.

[22] European Commission. Commission recommendation (EU) 2021/946 of 3 June 2021 on a common Union Toolbox for a coordinated approach towards a European Digital Identity Framework. https://eur-lex.europa.eu/eli/reco/2021/946, 2021. Official Journal of the European Union L 2021/946, 16 June 2021.

[23] European Commission. Commission implementing regulation (eu) 2025/848 of 6 may 2025 laying down rules for the application of regulation (eu) no 910/2014 of the european parliament and of the council as regards the registration of wallet-relying parties. https://eur-lex.europa.eu/eli/reg_impl/2025/848/oj/eng, 2025. Official Journal of the European Union L 2025/848, 7 May 2025.

[24] European Commission. Commission releases enhanced second version of the age-verification blueprint, oct 2025. URL: https://digital-strategy.ec.europa.eu/en/policies/eu-age-verification.

[25] European Commission. Eu age verification solution, 2025. https://ageverification.dev/.

[26] European Commission. Age verification use case manual, 2026. https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/930450954/The+Age+Verification+Manual.

[27] European Commission. EU Digital Identity Use Case Manuals, 2026. URL: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/896827987/Use+case+manuals.

[28] European Commission. The e-signature Manual, 2026. Accessed 20-03-2026. URL: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/930452287/eSignature.

[29] European Commission. The identification in proximity scenarios Manual, 2026. Accessed 20-03-2026. URL: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/930451396/Identification+in+proximity+scenarios.

[30] European Commission. The Payment Authentication Manual, 2026. Accessed 24-03-2026. URL: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/935397429/Payment+Authentication.

[31] European Commission. The PID-based identification for online services Manual, 2026. Accessed 20-03-2026. URL: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/930451131/PID+Identification+Manual.

[32] European Digital Identity Cooperation Group and European Commission. European digital identity wallet architecture and reference framework; v2.7.3. https://eudi.dev/2.7.3/architecture-and-reference-framework-main/, 2023.

[33] European Digital Identity Cooperation Group and European Commission. European digital identity wallet architecture and reference framework — discussion topics; v2.8.0. https://eudi.dev/2.8.0/discussion-topics/, 2026.

[34] European Digital Identity Cooperation Group and European Commission. European digital identity wallet architecture and reference framework; v2.8.0. https://eudi.dev/2.8.0/architecture-and-reference-framework-main/, 2026.

[35] European Digital Identity Cooperation Group and European Commission. European digital identity wallet pilot projects. https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/694487808/What+are+the+Large+Scale+Pilot+Projects, 2026.

[36] European Digital Identity Cooperation Group and European Commission. European digital identity wallet reference implementation. https://github.com/eu-digital-identity-wallet/.github/blob/main/profile/reference-implementation.md, 2026.

[37] European Digital Rights (EDRi). Online age verification and children's rights, 10 2023. https://edri.org/wp-content/uploads/2023/10/Online-age-verification-and-childrens-rights-EDRi-position-paper.pdf.

[38] European Parliament and Council of the European Union. 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. https://eur-lex.europa.eu/eli/reg/2014/910/oj/eng, 2014. Official Journal of the European Union L 257, 28.8.2014, pp. 73–114. URL: https://eur-lex.europa.eu/eli/reg/2014/910/oj/eng.

[39] European Parliament and Council of the European Union. Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 april 2024 amending Regulation (EU) no 910/2014 as regards establishing the European Digital Identity Framework. https://eur-lex.europa.eu/eli/reg/2024/1183/oj/eng, 2024. Official Journal of the European Union L 2024/1183, 30 April 2024; in force.

[40] European Parliament and Council of the European Union. 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 (consolidated version: 18 october 2024). https://eur-lex.europa.eu/eli/reg/2014/910/2024-10-18, 2024. Consolidated text incorporating amendments up to 18 October 2024; original OJ L 257, 28.8.2014, pp. 73–114.

[41] International Organization for Standardization. Personal identification — iso-compliant driving licence. part 5: Mobile driving licence (mdl) application, 2021. ISO/IEC 18013–5:2021.

[42] Karla Friedrichs, Franklin Harding, Anja Lehmann, and Anna Lysyanskaya. Device-bound anonymous credentials with(out) trusted hardware. Cryptology ePrint Archive, Paper 2025/1995, 2025. URL: https://eprint.iacr.org/2025/1995.

[43] Matteo Frigo and abhi shelat. Anonymous credentials from ECDSA. Cryptology ePrint Archive, Paper 2024/2010, 2024. URL: https://eprint.iacr.org/2024/2010.

[44] Liza Gak, Seyi Olojo, and Niloufar Salehi. The distressing ads that persist: Uncovering the harms of targeted weight-loss ads among users with histories of disordered eating. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2), November 2022. doi:10.1145/3555102.

[45] Francesco Gazzetta. Discussion comment on feedback #577 on eudif. https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/issues/577#issuecomment-3140483427, 2025.

[46] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering privacy by design. In *Computers, Privacy & Data Protection (CPDP)*, 2011.

[47] Marian Harbach, Sascha Fahl, Matthias Rieger, and Matthew Smith. On the acceptance of privacy-preserving authentication technology: The curious case of national identity cards. In Emiliano De Cristofaro and Matthew K. Wright, editors, *PETS 2013*, volume 7981 of *LNCS*, pages 245–264. Springer, Berlin, Heidelberg, July 2013. doi:10.1007/978-3-642-39077-7_13.

[48] Connor Jones. Signal will withdraw from Sweden if encryption-busting laws take effect. *The Register*, February 2025.

[49] Jason Kelley and Adam Schwartz. Age verification mandates would undermine anonymity online. Electronic Frontier Foundation, 3 2023. https://www.eff.org/deeplinks/2023/03/age-verification-mandates-would-undermine-anonymity-online.

[50] Merel Koning, Paulan Korenhof, Gergely Alpár, and Jaap-Henk Hoepman. The ABCs of ABCs: an analysis of attribute-based credentials in the light of data protection, privacy and identity. In Joan Balcells Padullés, Agustí Cerrillo i Martínez, Miquel Peguera Poch, Ismael Peña López, María José Pifarré de Moner, and Mònica Vilasau Solana, editors, *Internet, Law & Politics : A decade of transformations*, volume 10, pages 357–374, Barcelona, 2014. Huygens Editorial.

[51] René Mayrhofer, Anja Lehmann, and abhi shelat. A brief note on cryptographic pseudonyms for anonymous credentials, 2025. URL: https://arxiv.org/abs/2510.05419, arXiv:2510.05419.

[52] Zahra Moti, Asuman Senol, Hamid Bostani, Frederik J. Zuiderveen Borgesius, Veelasha Moonsamy, Arunesh Mathur, and Gunes Acar. Targeted and troublesome: Tracking and advertising on children's websites. In *2024 IEEE Symposium*

on Security and Privacy, pages 1517–1535. IEEE Computer Society Press, May 2024. doi:10.1109/SP54263.2024.00118.
[53] No Phone Home. No phone home, May 2025. https://nophonehome.com/.
[54] Republic of Estonia. Identity documents act–riigi teataja. https://www.riigiteataja.ee/en/eli/504022020003/consolide, April 2020.
[55] Unique Identification Authority of India. Vision & mission - unique identification authority of india. https://uidai.gov.in/en/about-uidai/unique-identification-authority-of-india/vision-mission.html, January 2026.
[56] Office of the United Nations High Commissioner for Human Rights. The struggle of trans and gender-diverse persons, 2026. April 2026. URL: https://www.ohchr.org/en/special-procedures/ie-sexual-orientation-and-gender-identity/struggle-trans-and-gender-diverse-persons.
[57] Christian Paquin, Guru-Vamsi Policharla, and Greg Zaverucha. Crescent: Stronger privacy for existing credentials. Cryptology ePrint Archive, Paper 2024/2013, 2024. URL: https://eprint.iacr.org/2024/2013.
[58] Parliament of Australia. Online safety amendment (social media minimum age) bill 2024, 2024. https://www.aph.gov.au/Parliamentary_Business/Bills_Legislation/Bills_Search_Results/Result?bId=r7284.
[59] Jeannie Marie Paterson, Shanton Chang, Marc Cheong, Chris Culnane, Suelette Dreyfus, and Dana McKay. The hidden harms of targeted advertising by algorithm and interventions from the consumer protection toolkit. *International Journal on Consumer Law and Practice*, 9, 2021.
[60] Singapore Government Developer Portal. Verify. https://www.developer.tech.gov.sg/products/categories/digital-identity/verify/overview, December 2025.
[61] Jae A Puckett, Nina C Brown, Terra Dunn, Brian Mustanski, and Michael E Newcomb. Perspectives from transgender and gender diverse people on how to ask about gender. *LGBT health*, 7(6):305–311, 2020.
[62] Imran Rahman-Jones. Pornhub leaves France over age verification law. *BBC*, June 2025.
[63] Ahmad Sabouri. On the user acceptance of privacy-preserving attribute-based credentials - A qualitative study. In *DPM/QASAESORICS*, volume 9963 of *Lecture Notes in Computer Science*, pages 130–145, Cham, 2016. Springer.
[64] Bruce Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal*, 24(12):21–29, 1999.
[65] S. Shanza, Ameemah Humayun, Urs Hengartner, and Leah Zhang-Kennedy. Sharing digital devices is normal and cultural: Privacy and security challenges in collectivist immigrant households. In *Twenty-Second Symposium on Usable Privacy and Security (SOUPS 2026)*, pages 117–139. USENIX Association, August 2026. URL: https://www.usenix.org/conference/soups2026/presentation/shanza.
[66] Jay Stanley. Identity crisis: What digital driver's licenses could mean for privacy, equity, and freedom. ACLU, 5 2021. https://www.aclu.org/publications/identity-crisis-what-digital-drivers-licenses-could-mean-privacy-equity-and-freedom.
[67] Erik Wästlund, Julio Angulo, and Simone Fischer-Hübner. Evoking comprehensive mental models of anonymous credentials. In *Proceedings of the 2011 IFIP WG 11.4 International Conference on Open Problems in Network Security*, pages 1–14, Berlin, Heidelberg, 2011. Springer-Verlag.
[68] Yuxi Wu, Sydney Bice, W. Keith Edwards, and Sauvik Das. The slow violence of surveillance capitalism: How online behavioral advertising harms people. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, pages 1826–1837, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3593013.3594119.

A A Taxonomy of Harms for DIFs

Economic Harms involve financial losses or loss in the value of something. Use of the EUDIF for instant payments, account-to-account transfers, and other cross-border digital transactions is recognized as one of its primary use cases. The EUDIF thus brings with it a large potential to cause economic harms due to unintended inference / leakage of private information. The potential economic harms of the EUDIF include:

- Financial loss through impersonation,
- Financial loss because blocked from access to own bank account,
- Burglary at one's place of residence.

Physical Harms are those that result in any form of injury or death. Albeit a "digital system", misuse or improper protection of the DIF might lead to physical harms such as:

- Physical violence through individual aggressor (stalker learns place of residence),
- Decrease in personal health due to insufficient access to health care,
- Loss of access to critical commodities like energy/heating at home because suddenly requires online verification/registration.

Psychological Harms are any kind of negative mental response such as emotional distress or disturbance. This includes a wide array of feelings that can be triggered by privacy violations such as anxiety, anguish, concern, or irritation. In the context of DIFs, we can identify the following psychological harms:

- Fear of being watched in everyday activities,
- Emotional distress due to identity theft,
- Fear of traveling/crossing borders because might be surveiled.

Autonomy/Deprivation Harms consider harms resulting from restricting, undermining, inhibiting, or unduly influencing individuals' choices, such as

- Coercion,
- Manipulation,
- Restriction or deprivation of individual freedom of speech,
- Restriction or deprivation of individual freedom of information,
- Restriction or deprivation of freedom of movement.

Discrimination Harms based on demographic attributes:

- Profile-based discrimination
- Dynamic pricing (airlines, ski pass, hotels, etc.).

Reputational Harms are harms to reputation or dignity. They are particularly relevant in the context of DIFs, as DIFs generate information about what users do and which services they use.

Societal Harms include chilling effect or loss of creativity affecting the society due to constant monitoring by government or law enforcement bodies. DIFs may lead to severe societal harms:

- Impairments on freedom of speech,
- Impairments on free elections and other democratic processes.

B Body of Law

The body of law and specifications for the European Digital Identity Framework and Wallet consist of the following:

- Consolidated EU regulation [40] of the base regulation (EU) No 910/2014 [38] (eIDAS) amended by Regulation (EU) 2024/1183 [39];
- Implementing acts:
  - Implementing Regulation (EU) 2024/2977: personal identification data and electronic attestations of attributes;
  - Implementing Regulation (EU) 2024/2979: integrity and core functionalities of European Digital Identity Wallets;
  - Implementing Regulation (EU) 2024/2980: trust framework;
  - Implementing Regulation (EU) 2024/2981: certification of European Digital Identity Wallets;
  - Implementing Regulation (EU) 2024/2982: protocols and interfaces to be supported;
  - Implementing Regulation (EU) 2025/846: cross-border identity matching for natural persons;
  - Implementing Regulation (EU) 2025/847: reactions to security breaches of wallets;
  - Implementing Regulation (EU) 2025/848: registration of wallet-relying parties [23];
  - Implementing Regulation (EU) 2025/849: submitting information to the Commission for the list of certified wallets;

  - Implementing Regulation (EU) 2025/1566: verification of identity and attributes when issuing qualified certificates or qualified electronic attestations.
  - Implementing Regulation (EU) 2025/1567: management of remote qualified electronic signature/seal creation devices;
  - Implementing Regulation (EU) 2025/1568: procedural arrangements for peer reviews of eID schemes;
  - Implementing Regulation (EU) 2025/1569: electronic attestations of attributes (EAA/QEAA specifications);
  - Implementing Regulation (EU) 2025/1570: notifications of certified qualified signature/seal creation devices;
  - Implementing Regulation (EU) 2025/1572: initiation of qualified trust services (notification of intention);
  - Implementing Regulation (EU) 2025/1571: annual reports from supervisory bodies;
- Supporting governance/technical documentation (EUDI Architecture and Reference Framework (ARF) [32]);
- Adjacent EU laws (e.g., GDPR, accreditation, cybersecurity).

C EUDIF Use Cases

We review the EUDIF use cases for which there exists documentation of their intended use [27], and summarize the use cases in Table 1.

The scenarios describing generic applications of the framework for identification purposes are the *PID-based identification for online services* [31] and *identification in proximity scenarios* [29] use cases. The former is intended for online service providers to verify the personal identification data (PID, i.e., full name, date of birth, nationality, and other details) of the user via the EUDIF. The latter is similarly for the purpose of identifying the user, but in proximity scenarios, i.e., for face-to-face interactions. The *e-signature* use case [28] describes a scenario where an individual wishes to sign a contract (e.g., employment contract, rental contract). In this use case, a third party verifies the individual's identity and signs documents on their behalf. The *payment authentication* use case [30] functions similarly: a third-party payment service provider verifies that an individual wishes to make a payment (and their identity), and executes the transaction for the user. The remaining use cases are either identity use cases that result in access to services or freedoms (e.g., Mobile Driving License, Digital Travel Credential, ePrescription, and European Health Insurance Card), or non-identity use cases that result in access to services or freedoms (e.g., Age Verification, European Parking Card, European Disability Card).

We briefly discuss how the EUDIF use cases (beyond age verification and mobile driving licences) can be formally captured by our modeling. Use cases that result in access to services or spaces are captured similarly to the functionalities for age verification and mDLs. Generic identity cases (PID-based identification for online services and identification in proximity scenarios) are directly captured by $\mathcal{F}_{\mathsf{EUDIF}}$ when the issuer is an entity with the authority to issue identity information, the RP is the server provider, and the output returned to the RP is the user's identity information.

Some use cases involve a third-party entity that performs credential verification and forwards this information to a service provider. In our modelling, this third-party acts as the RP. For example, the *e-signature* use case [28] describes a generic application of the EUDIF where a third-party "secure trust provider" signs documents on behalf of an individual, whose identity they must verify. Here, the RP in our model is the "secure trust provider." The signed document can then be sent to the entity who generated the contract. The "payment service provider" of the *payment authentication* use case [30] similarly acts as RP: it verifies a user's proof of payment authentication and identity, and then executes the transaction. The merchant then receives a confirmation from the payment service provider. We note that our modelling, where we define the RP to be the entity performing verification of attributes, differs from the use cases' documentation which refers to merchants as the RP.

D Functionality for Mobile Driving License

In this use case, RPs are car rental services, who register with a registration authority by calling the *RP registration* subroutine of $\mathcal{F}_{\mathsf{EUDIF}}$. Users are individuals accessing car rental services, and can request the issuance of an identity credential and an mDL credential from a government (or similar) entity $\mathcal{I}_{\mathsf{iid}}$, by calling the *Issuance* subroutine of $\mathcal{F}_{\mathsf{EUDIF}}$.

To access the service via the EUDIF functionality, we require that the car rental service is registered with an authority as a RP (RegRP[rid].status = registered) and that the user has been issued an mDL and identity credentials.

When visiting a car rental service (in person), the clerk requests identification of the user. For example, a user may open their EUDI wallet application to engage with the verifier by scanning a QR code that requests verified information from them, including their identity and mDL credentials. This is captured in the underlying $\mathcal{F}_{\mathsf{EUDIF}}$ functionality, in *Presentation* (4) where the user $\mathcal{U}_{\mathsf{uid}}$ obtains information about the RP, a list of desired attributes, the intended use, and the function to be computed over the attributes f.

The user reviews the request to present their data, and determines (i) whether they first check the registration status of the RP (captured by O_{check}); and (ii) whether they consent to engaging with the system and providing this information (captured by O_{eng}). Upon confirmation of disclosure, the verifier receives the information and validates the proof locally. Here, the function f determines if the user has the right to drive the desired vehicle class. Additionally, the function f returns to the verifier the identity information. For example, a verifier device will show the identity credential's associated photo to the clerk, who can verify the user's identity. This out-of-band verification is captured by the oracle O_{ver}.

Then, the verifier device validates the proof locally and shows the document's picture to the clerk who verifies the user identity. This out-of-band verification, and the clerk's subsequent decision to grant access to the user based on the output of f, is captured by the oracle O_{ver}. If access is granted, the clerk continues with check in of the user for their car rental.

E Additional Harm Trees

Here, we present additional harm trees related to the age verification use case.

We first describe a sub-tree of a capability that is common to several of the following harms. Requiring users to prove their age to a service provider (to access the service) or requiring them to

Table 1: EUDIF Use Cases. "Identity" refers to whether or not it is an identity use case (i.e., identity is revealed to the relying party). "Relying Party" and "Issuer" columns correspond to how these entities are modeled in $\mathcal{F}_{\text{EUDIF}}$, i.e., the RP is the entity that receives the output of some function evaluated on the relevant attributes. "(Off/On)line" indicates whether the use case is targeted primarily towards offline or online scenarios or both.

Use Case	Relevant Attributes	Identity	Off/Online	Issuer	Relying Party	Result
Generic Use Cases						
PID-based identification for online services	Identity	✓	Online	Government (or similar)	Various	Various
Identification in proximity scenarios	Identity	✓	Offline	Government (or similar)	Various	Various
Specific Use Cases						
eSignature	Identity	✓	Both	Government (or similar)	Secure Trust Provider	Document signed and forwarded
Mobile Driving License	Driving rights, identity	✓	Both	Government	Car Rental Service; Law Enforcement	Access to car rental; Roadside checks passed
Payment Authentication	Payment amount, recipient, identity, other (e.g., age, address)	✓	Both	Payment Service Provider	Payment Service Provider	Executes transaction and merchant receives confirmation
Age Verification	Age	✗	Both	Government (or similar)	Various (e.g., adult content sites, gambling platforms)	Access to services
Digital Travel Credential	Travel document (e.g., passport), travel info (e.g., flight number)	✓	Offline	Government	Border Authority	Border checks passed
European Parking Card	Parking access rights	✗	Offline	Various	Parking Authority	Parking inspection checks passed
European Disability Card	Disability status	✗	Both	Government	Various (e.g., museum or transit services)	Access to services and discounts
ePrescription	Identity (Health ID)	✓	Offline	Government	Pharmacies	Access to medication
European Health Insurance Card	Insurance status, identity	✓	Offline	Government	Healthcare providers	Access to healthcare

access a device may result in users choosing *not* to engage with the service (Fig. 10). The requirement for access to a device arises due to digitalization and the requirement for users to prove their age is inherent to any EUDIF system. Users may opt not to engage for fear of tracking and surveillance, due to distrust in the service provider, or because they do not want to reveal their age information. This decision of the user is captured by the oracle O_{eng}.

Reduced availability of resources. The introduction of the EUDIF might lead to a chilling effect on legitimate activities. Less information might be published because of the additional burden on operators to implement age verification. Individuals may refrain from publishing content online if they must first prove their age. Service providers might pull out of territories due to the additional burden of compliance; this results in fewer services being available. The paths to this harm are inherent to any system implementing age verification for online services, illustrated in Fig. 11.

Censorship. The EUDIF system, when used for age verification, can lead to censorship in multiple ways. A central entity (in our modeling, CA) is given the power to decide which services require age verification. This central entity can restrict what children can see online and which platforms and services they can use. Thus ultimately impairs children's self-development (due to biased educational content) and restricts their freedom of expression (by blocking access to private messengers or social media). Furthermore, function creep/re-purposing of the system is enabled by the central entity deciding which content requires age verification or low-integrity operators/service providers choosing whether or not to implement age verification. In our modeling, the operator can opt to register (by sending Reg, and based on the decision of O_{reg}) regardless of whether the CA requires that they do so. This captures that once the infrastructure for widespread age verification is in place, any service provider may choose to integrate into the system. The EUDIF system also leads to some forms of self-censorship. Individuals might refrain from publishing content online if it requires using the system to prove their age, undermining freedom of expression. The resulting censorship harm (either via self-censorship, or imposed by the operators or central entity) is inherent to the functionality of the use case.

Diminished Internet safety. The EUDIF bears potential for tracking and de-anonymising online users and relies on an authority to issue a proof of age to access certain services. As a result, it prevents anyone who needs independence from the issuing authority from

Parties. Issuer $\mathcal{I}_{\mathsf{DL}}$ is a national government (or similar) entity that issues driving license and identity credentials to users. Users $\mathcal{U}_{\mathsf{uid}}$ are individuals accessing car rental services. Relying Parties $\mathcal{R}_{\mathsf{rid}}$ are car rental services.
Parameters. Let f be a function that takes as input, a user's mDL attribute mDL and their identity attribute identity, and a vehicle class vc. It returns (i) a boolean whether the user has the right to drive vehicle class vc in the current jurisdiction *and* their mDL credential is currently valid, and (ii) additional identity information: the user's photo, name, date of birth, and nationality. Let $\mathcal{O}_{\mathsf{reg}}$ be an oracle that upon input rid outputs a bit $b \in \{0,1\}$ determining whether $\mathcal{R}_{\mathsf{rid}}$ registers as a relying party. Let $\mathcal{O}_{\mathsf{check}}$ be an oracle that upon input $(\mathsf{rid}, \mathsf{intended_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, \mathsf{uid})$, outputs a bit $b \in \{0,1\}$ determining whether $\mathcal{U}_{\mathsf{uid}}$ checks $\mathcal{R}_{\mathsf{rid}}$'s registered information. Let $\mathcal{O}_{\mathsf{eng}}$ be an oracle that upon input $(b_{\mathsf{reg}}, \mathsf{uid})$ or $(b_{\mathsf{pol}}, \mathsf{uid})$ outputs a (consent) bit $\mathsf{consent}_1, \mathsf{consent}_2 \in \{0,1\}$ determining whether $\mathcal{U}_{\mathsf{uid}}$ consents to engage with the system. Let $\mathcal{O}_{\mathsf{ver}}$ be an oracle that upon input (v, rid), outputs a bit $b \in \{0,1\}$ that determines whether the RP $\mathcal{R}_{\mathsf{rid}}$ grants access to the user, based on the output v of f.
RP Registration. Send $(\mathsf{RegRP}, \mathsf{driving\text{-}rights}, f, (\mathsf{mDL}, \mathsf{identity}))$ from $\mathcal{R}_{\mathsf{rid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$.
Request mDL issuance. On input $(\mathsf{RequestIssuance}, \mathsf{uid})$ from $\mathcal{U}_{\mathsf{uid}}$:
(1) Send $(\mathsf{Issue}, \mathsf{DL}, \mathsf{identity} = (\mathsf{name}, \mathsf{nationality}, \mathsf{photo}, \mathsf{dob}))$ from $\mathcal{U}_{\mathsf{uid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$.
(2) Send $(\mathsf{Issue}, \mathsf{DL}, \mathsf{mDL} = (\mathsf{driving\text{-}rights}))$ from $\mathcal{U}_{\mathsf{uid}}$ to $\mathcal{F}_{\mathsf{EUDIF}}$ and obtain o in response.
Access car rental. On input $(\mathsf{AccessCarRental}, \mathsf{rid})$ from $\mathcal{U}_{\mathsf{uid}}$:
(1) Send $(\mathsf{RequestPresentation}, \mathsf{rid}, \mathsf{CarRental})$ from uid to $\mathcal{F}_{\mathsf{EUDIF}}$.
(2) Upon receipt of $(\mathsf{Req}, \mathsf{CarRental})$, $\mathcal{R}_{\mathsf{rid}}$ returns $(f, (\mathsf{identity}, \mathsf{mDL}), 1)$.
(3) Send $(\mathsf{driver\text{-}data}, f, \mathsf{mDL}, (\mathsf{identity}, \mathsf{mDL}))$ to $\mathcal{O}_{\mathsf{check}}$ (upon receipt from $\mathcal{F}_{\mathsf{EUDIF}}$) and return the result as b_{check}.
(4) Upon receipt of b_{reg} (resp., b_{pol}), $\mathcal{U}_{\mathsf{uid}}$ forwards the input to $\mathcal{O}_{\mathsf{eng}}$ and returns the result as $\mathsf{consent}_1$ (resp., $\mathsf{consent}_2$).
(5) Upon receipt of the result v (output by f) $\mathcal{R}_{\mathsf{rid}}$ gets $b \leftarrow \mathcal{O}_{\mathsf{ver}}(v, \mathsf{rid})$. If $b = 1$, return GrantAccess to $\mathcal{U}_{\mathsf{uid}}$; else return DenyAccess.

Figure 9: Mobile driving license for car rentals functionality $\mathcal{F}_{\mathsf{mDL}}[\mathcal{F}_{\mathsf{EUDIF}}]$.

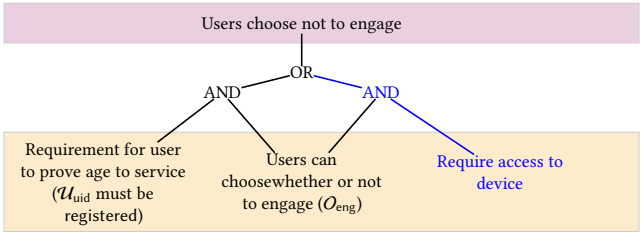


Figure 10: Age verification – Sub-tree for the capability of users choosing not to engage

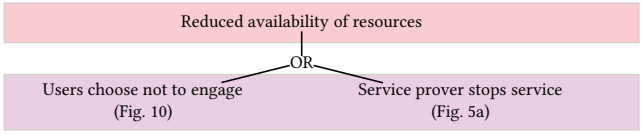


Figure 11: Age verification – Reduced availability

safely using the Internet. Users are discouraged from expressing unpopular opinions if they need to fear de-anonymisation. This is harmful because it erodes democratic freedoms such as freedom of expression and opinion.

Increased cybersecurity risks For similar reasons that an operator may opt to stop providing its service, it may opt to simply become non-compliant. That is, we can replace the capability in Fig. 5a with "Service provider becomes unregulated." Additionally, users may opt not to engage with the age verification system (Fig. 10). This suggests that users might be pushed to unsafe/unregulated spaces and the amount of such unregulated services will increase.

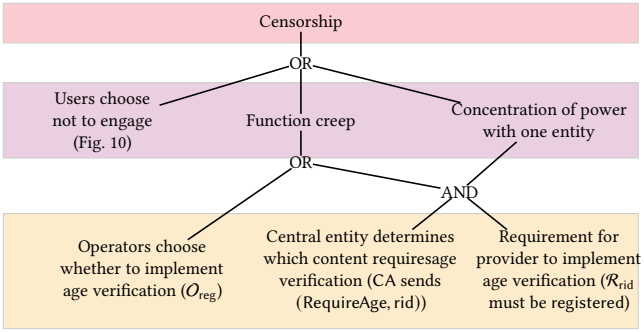


Figure 12: Age verification – Censorship

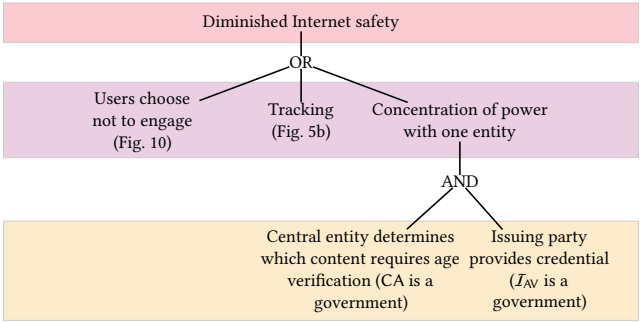


Figure 13: Age verification – Diminished Internet safety

A naive implementation exacerbates this harm even further, by

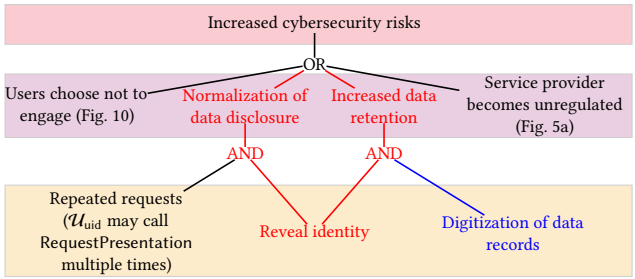


Figure 14: Age verification – Increased cybersecurity risks

normalizing requests for and increasing retention of, identity information. Altogether, this increases users' exposure to risks such as fraud or theft.

F Security Games

Security games capture the capabilities of an adversary – captured via oracles (auxiliary data, adaptive behaviour, number of queries, etc.) – and define the adversary's winning condition (i.e., a bad event that should not happen). An adversary's advantage is defined as the probability of winning the game minus a baseline winning probability (e.g., random guessing). An adversary is said to win the game if its advantage is non-negligible, i.e., they can trigger the bad event more often than the baseline suggests.

To test whether a functionality $\mathcal{F}$ satisfies a given property, we use *universally-composable-compatible security games* [7].[3]

Let $\mathcal{F}$ be a functionality, let Π be a protocol realizing $\mathcal{F}$, let $\mathcal{Z}$ be the environment, let $\mathcal{S}$ be the simulator (against $\mathcal{F}$), and let $\mathcal{A}$ be the adversary (against Π). In the ideal world, the challenger interacts with the ideal-world adversary (i.e., the simulator) $\mathcal{S}$. In the real world, the challenger interacts with the real-world adversary $\mathcal{A}$. We write $\mathcal{G}_{\mathcal{S},\mathcal{F}}$ for the ideal-world game, and $\mathcal{G}_{\mathcal{A},\Pi}$ for the real-world game. A game $\mathcal{G}_{\mathcal{A},\Pi}$ is defined by a challenger C with black-box access to the parties (i.e., it interacts with them through their environment-facing interfaces). C interacts with $\mathcal{A}$ as the environment, and outputs a bit b indicating whether the adversary wins the game.

We consider that an adversary wins if it can trigger a *bad event* for any one of all possible consent strategies. This mirrors both standard definitions of security games in cryptography, where one considers security against *worst-case* inputs. This choice of definition also captures the harms-based focus of our analysis: if the adversary can trigger a bad event for a specific consent strategy, then harms may materialize for any user with this consent strategy; for DIFs with many millions of envisioned users, this is a non-negligible number of users for which harms may materialize.

F.1 Distinguishing Game

In the distinguishing game presented in Fig. 15, the adversary's goal is to distinguish between two fresh challenge users. The game proceeds in two phases. In the challenge issuance phase, the adversary specifies which attributes and policies are set for these users. The challenger internally calls the $\mathcal{F}_{\text{EUDIF}}$ functionality to issue two credentials. Then, during the challenge presentation phase, the adversary, acting as a malicious RP, can interact with one of the two fresh challenge users. The adversary can specify both users' behaviour with respect to registration checking and consent. It does so by by specifying functions $B^{(i)}_{\text{reg}} : \{0,1\} \to \{0,1\}$ and $B^{(i)}_{\text{pol}} : \{0,1\} \to \{0,1\}$. These specify how the user respectively responds to $b_{\text{registered}}$ and b_{policy} bits. Again, the challenger uses $\mathcal{F}_{\text{EUDIF}}$ to create presentations.

The game models three variants, policy, attributes, consent, where the adversary is only allowed to induce differences in one of these three controls. We formalize our results.

We first look at distinguishing based on policy.

Lemma F.1 (Distinguishing based on policy is inherent to $\mathcal{F}_{\text{EUDIF}}$.). *There exists an adversary $\mathcal{A}_{\text{dist,policy}}$ that wins the discrimination game $\mathcal{G}^{\text{dist,policy}}$ against $\mathcal{F}_{\text{EUDIF}}$ with advantage* 1.

Proof sketch. We proceed by demonstrating an explicit win scenario. During the challenge issuance step, the adversary specifies one attribute, 'gender', and sets the values of users $\mathcal{U}_0$ and $\mathcal{U}_1$ to 'X'. However, for $\mathcal{U}_0$ it sets policy_0 to always return 0, while policy_1 always returns 1. At the challenge presentation step, the adversary requests disclosure of the 'gender' attribute. The adversary requests $B^{(0)}_{\text{pol.}} = B^{(1)}_{\text{pol.}}$ to be the identity function. Clearly, then user $\mathcal{U}_0$ aborts (due to policy check) and $\mathcal{U}_1$ proceeds.

Note that the conditions for win = policy are satisfied, and that the the adversary can perfectly distinguish between the two users. It thus wins $\mathcal{G}^{\text{dist,policy}}$ with advantage 1. □

We now look at distinguishing based on attributes. We state the lemma without a proof. The adversary simply issues different attribute values to the two challenge users, and requests to see this values during presentation. This approach leads to a perfect distinguishing advantage.

Lemma F.2 (Distinguishing based on attributes is inherent to $\mathcal{F}_{\text{EUDIF}}$.). *There exists an adversary $\mathcal{A}_{\text{dist,attribute}}$ that wins the distinguishing game $\mathcal{G}^{\text{dist,attribute}}$ against $\mathcal{F}_{\text{EUDIF}}$ with advantage* 1.

Finally, we look at the effect of user consent.

Lemma F.3 (Distinguishing based on consent is inherent to $\mathcal{F}_{\text{EUDIF}}$.). *There exists an adversary $\mathcal{A}_{\text{dist,consent}}$ that wins the distinguishing game $\mathcal{G}^{\text{dist,consent}}$ against $\mathcal{F}_{\text{EUDIF}}$ with advantage* 1.

Proof sketch. We proceed by demonstrating an explicit win scenario. At the challenge issuance step, the adversary specify one attribute, 'age', and sets the values of users $\mathcal{U}_0$ and $\mathcal{U}_1$ to '22', and sets $\text{policy}_0 = \text{policy}_1$ to always return 1. However, for $\mathcal{U}_0$, it sets the user to be privacy conscious and only consent if the RP has been registered (i.e., set $B^{(0)}_{\text{reg}}$ to be the identity function). For $\mathcal{U}_1$, it sets the user to be more lax and always consent (i.e., $B^{(1)}_{\text{reg}} = 1$). Clearly, for an unregistered RP, $\mathcal{U}_0$ aborts and $\mathcal{U}_1$ proceeds. We note that the conditions for win = consent are satisfied, and that the the adversary can perfectly distinguish between the two users. It thus wins $\mathcal{G}^{\text{dist,consent}}$ with advantage 1. □

[3] Note that we do not use universal composability (UC) in this work, but merely the notion of games defined against ideal-world functionalities.

Setup.
(1) $\mathcal{A}$ selects a corrupted relying party $\tilde{\mathcal{R}}$.
(2) C creates an issuer $\mathcal{I}$ and users $\mathcal{U}_0, \mathcal{U}_1$, and initializes an instance of $\mathcal{F}_{\mathsf{EUDIF}}$ with issuers $\{\mathcal{I}\}$, users $\{\mathcal{U}_0, \mathcal{U}_1\}$ and RPs $\{\tilde{\mathcal{R}}\}$.
Challenge issuance. $\mathcal{A}$ instructs challenger C to issue a credential to challenge users $\mathcal{U}_0$ and $\mathcal{U}_1$. $\mathcal{A}$ sends inputs $(\mathtt{ChalIss}, \mathsf{iid}, (\mathsf{attr_descr}_i)_{i\in[n]}, (\mathsf{attr}_i^{(0)})_{i\in[n]}, (\mathsf{attr}_i^{(1)})_{i\in[n]}, \mathsf{policy}_0, \mathsf{policy}_1)$ to C. The challenger sends two issuance requests to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of users $\mathcal{U}_0$ and $\mathcal{U}_1$ for the issuer iid and the attribute types $(\mathsf{attr_descr}_i)_{i\in[n]}$. For user $\mathcal{U}_j$, $j \in \{0,1\}$, C programs the issuance oracle to return the attributes $(\mathsf{attr}_i^{(j)})_{i\in[n]}$ and the disclosure policy policy_j.
Challenge presentation. $\mathcal{A}$ may instruct C to request – for one of the two challenge users – access for an adversary-specified service for the adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, \mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, B_{\mathsf{reg}}^{(0)}, B_{\mathsf{reg}}^{(1)}, B_{\mathsf{pol}}^{(0)}, B_{\mathsf{pol}}^{(1)})$ to C. C then interacts with $\mathcal{F}_{\mathsf{EUDIF}}$ as follows. It sends a presentation request $(\mathsf{Req}, \mathsf{rid}, \mathsf{service})$ to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}_b$. In response to the $(\mathsf{Request}, \mathsf{service})$ message for $\tilde{\mathcal{R}}$, it responds with $(\mathsf{aux} = (\mathsf{rp_info}, \mathsf{int_use}), f, (\mathsf{attr_descr}_i)_{i\in[n]})$. On input $(\mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ it sends $b_{\mathsf{check}} = 1$ (w.l.o.g, we assume users always check). On input $b_{\mathsf{registration}}$, the challenger returns $B_{\mathsf{reg}}^{(b)}(b_{\mathsf{registration}})$. On input b_{policy}, the challenger returns $B_{\mathsf{pol}}^{(b)}(b_{\mathsf{policy}})$. Finally, the challenger returns the response v (which may equal $\perp$) to the adversary.
Modeling strategies. We capture three winning strategies:
- win = policy: enforce $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$, and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$, otherwise the game returns 0.
- win = attributes: enforce $\mathsf{policy}_0 = \mathsf{policy}_1$, $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, and $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$, otherwise the game returns 0.
- win = consent: enforce $\mathsf{policy}_0 = \mathsf{policy}_1$ and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$, otherwise the game returns 0.

Winning. $\mathcal{A}$ outputs a bit b'. The game outputs 1 iff $b = b'$.
$\mathcal{A}$'s advantage in the game is $\mathsf{Adv}^{\mathsf{dist,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}}} := 2 \cdot \left|\Pr\left[\mathcal{G}^{\mathsf{dist,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b} \to 1\right] - \frac{1}{2}\right| \in [0,1]$.

Figure 15: Distinguishing game $\mathcal{G}^{\mathsf{dist,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b}$ for a challenge bit b, and winning condition $\mathsf{win} \in \{\mathsf{policy}, \mathsf{attributes}, \mathsf{consent}\}$.

F.2 Tracking Game

We describe two flavours of tracking: multi-show tracking and issuer tracking.

F.2.1 Multi-show tracking. In the multi-show tracking game (Fig. 16), the adversary's goal is to link two presentation of the same user to each other. The game is an extension of the distinguishing game (Fig. 15), where the adversary may additionally corrupt multiple relying parties, and observe multiple presentations of both challenge users. As for distinguishing, we consider three variants of the game, where the adversary can only induce differences in one of the three controls (policy, attributes, consent).

F.2.2 Issuer tracking. In the issuer tracking game (Fig. 17), the adversary's goal is to link a presentation of a user to the issuing event for attributes of this user. Compared to the distinguishing game, the adversary may additionally corrupt an issuer, and observe the issuing event for one of the two challenge users. As for distinguishing, we consider three variants of the game, where the adversary can only induce differences in one of the three controls (policy, attributes, consent).

F.2.3 Attack strategies. Multi-show and issuer tracking extend the distinguishing game by granting the adversary additional powers. Thus, any attacker successful in the distinguishing game is also successful in the tracking games (in particular, the adversaries described in App. F.1), which we can more formally state as follows.

Lemma F.4 (Tracking based on policy, attributes, or consent is inherent to $\mathcal{F}_{\mathsf{EUDIF}}$.). *There exist adversaries $\mathcal{A}_{\mathsf{track,policy}}$, $\mathcal{A}_{\mathsf{track,attribute}}$, and $\mathcal{A}_{\mathsf{track,consent}}$ that win the multi-show tracking game $\mathcal{G}^{\mathsf{track,multi-show}}$ and the issuer tracking game $\mathcal{G}^{\mathsf{track,issuer}}$ against $\mathcal{F}_{\mathsf{EUDIF}}$ with advantage 1.*

F.3 Inference Game

In the inference game (Fig. 18), the adversary's goal is to infer some information about a user's attributes. The game proceeds in two phases. In the challenge issuance phase, the adversary specifies which attributes and policies are set for a fresh challenge user. The challenger internally calls the $\mathcal{F}_{\mathsf{EUDIF}}$ functionality to issue a credential. Then, during the challenge presentation phase, the adversary, acting as a malicious RP, can interact with the fresh challenge user.

Quantifying inference. To quantify the inference capabilities of an adversary, we borrow ideas from the quantified information flows (QIF) literature [5].
- learn = full corresponds to the adversary learning the full attribute vector, and the advantage is the min-entropy loss (or equivalently, the increase in the adversary's guessing probability over the baseline where it knows only the distribution of attributes).
- learn = sensitive$(\overline{(\mathsf{attr_descr}_i)_{i\in[\ell]}})$ corresponds to the adversary learning only a subset of the attributes (considered sensitive), and the advantage is the min-entropy loss on this subset of attributes.

Setup.
(1) $\mathcal{A}$ selects a set of corrupted relying parties $\tilde{R}$.
(2) C creates an issuer $\mathcal{I}$ and users $\mathcal{U}_0, \mathcal{U}_1$, and initializes an instance of $\mathcal{F}_{\mathsf{EUDIF}}$ with issuers $\{\mathcal{I}\}$, users $\{\mathcal{U}_0, \mathcal{U}_1\}$ and RPs $\{\tilde{R}\}$.
Challenge issuance. $\mathcal{A}$ may instruct the challenge user $\mathcal{U}^*$ to request issuance of credentials, in a left-or-right fashion: $\mathcal{A}$ sends the message $(\mathtt{ChalIss}, \mathsf{iid}, (\mathsf{attr_descr}_i)_{i\in[n]}, (\mathsf{attr}_i^{(0)})_{i\in[n]}, (\mathsf{attr}_i^{(1)})_{i\in[n]}, \mathsf{policy})$ to C, with the restriction that $((\mathsf{attr}_i^{(0)})_{i\in[n]}, \mathsf{policy}_0) \neq ((\mathsf{attr}_i^{(1)})_{i\in[n]}, \mathsf{policy}_1)$. C then sends an issuance request to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}^*$ for the issuer iid and the attribute types $(\mathsf{attr_descr}_i)_{i\in[n]}$, and programs the issuance oracle to return the attributes $(\mathsf{attr}_i^{(b)})_{i\in[n]}$ and the disclosure policy policy.
Challenge presentation. $\mathcal{A}$ may instruct the challenge user $\mathcal{U}^*$ to request access for an adversary-specified service for an adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, B_{\mathsf{reg}}, B_{\mathsf{pol}.})$ to C. C then sends a presentation request to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}^*$ for the relying party rid and the service service, using the functions B_{reg} and B_{pol} to select the bits b_{reg} and b_{pol} that $\mathcal{U}^*$ sends to $\mathcal{F}_{\mathsf{EUDIF}}$ as part of the presentation interaction.
Challenge issuance. $\mathcal{A}$ instructs challenger C to issue a credential to challenge users $\mathcal{U}_0$ and $\mathcal{U}_1$. $\mathcal{A}$ sends inputs $(\mathtt{ChalIss}, \mathsf{iid}, (\mathsf{attr_descr}_i)_{i\in[n]}, (\mathsf{attr}_i^{(0)})_{i\in[n]}, (\mathsf{attr}_i^{(1)})_{i\in[n]}, \mathsf{policy}_0, \mathsf{policy}_1)$ to C. The challenger sends two issuance requests to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of users $\mathcal{U}_0$ and $\mathcal{U}_1$ for the issuer iid and the attribute types $(\mathsf{attr_descr}_i)_{i\in[n]}$. For user $\mathcal{U}_j$, $j \in \{0,1\}$, C programs the issuance oracle to return the attributes $(\mathsf{attr}_i^{(j)})_{i\in[n]}$ and the disclosure policy policy_j.
Presentations. $\mathcal{A}$ may instruct C to request – for either of the two challenge users – access for an adversary-specified service for the adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, \mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, B_{\mathsf{reg}}, B_{\mathsf{pol}}, j)$ to C. C then interacts with $\mathcal{F}_{\mathsf{EUDIF}}$ as follows. It sends a presentation request $(\mathsf{Req}, \mathsf{rid}, \mathsf{service})$ to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}_j$. In response to the $(\mathsf{Request}, \mathsf{service})$ message for $\tilde{\mathcal{R}}$, it responds with $(\mathsf{aux} = (\mathsf{rp_info}, \mathsf{int_use}), f, (\mathsf{attr_descr}_i)_{i\in[n]})$. On input $(\mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ it sends $b_{\mathsf{check}} = 1$ (w.l.o.g, we assume users always check). On input $b_{\mathsf{registration}}$, the challenger returns $B_{\mathsf{reg}}(b_{\mathsf{registration}})$. On input b_{policy}, the challenger returns $B_{\mathsf{pol}}(b_{\mathsf{policy}})$. Finally, the challenger returns the response v (which may equal $\perp$) to the adversary.
Challenge presentation. $\mathcal{A}$ may instruct C to request – for one of the two challenge users – access for an adversary-specified service for the adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, \mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, B_{\mathsf{reg}}^{(0)}, B_{\mathsf{reg}}^{(1)}, B_{\mathsf{pol}}^{(0)}, B_{\mathsf{pol}}^{(1)})$ to C. C then interacts with $\mathcal{F}_{\mathsf{EUDIF}}$ as follows. It sends a presentation request $(\mathsf{Req}, \mathsf{rid}, \mathsf{service})$ to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}_b$. In response to the $(\mathsf{Request}, \mathsf{service})$ message for $\tilde{\mathcal{R}}$, it responds with $(\mathsf{aux} = (\mathsf{rp_info}, \mathsf{int_use}), f, (\mathsf{attr_descr}_i)_{i\in[n]})$. On input $(\mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ it sends $b_{\mathsf{check}} = 1$ (w.l.o.g, we assume users always check). On input $b_{\mathsf{registration}}$, the challenger returns $B_{\mathsf{reg}}^{(b)}(b_{\mathsf{registration}})$. On input b_{policy}, the challenger returns $B_{\mathsf{pol}}^{(b)}(b_{\mathsf{policy}})$. Finally, the challenger returns the response v (which may equal $\perp$) to the adversary.
Modeling strategies. We capture three winning strategies:

- win = policy; Enforce $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$, and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$; otherwise the game returns 0.
- win = attributes; Enforce $\mathsf{policy}_0 = \mathsf{policy}_1$, $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, and $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$; otherwise the game returns 0.
- win = consent; Enforce $\mathsf{policy}_0 = \mathsf{policy}_1$ and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$; otherwise the game returns 0.

Guess. $\mathcal{A}$ outputs a bit b'. The game outputs 1 iff $b = b'$.
$\mathcal{A}$'s advantage in the game is $\mathsf{Adv}^{\text{ms-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}}} := 2 \cdot \left|\Pr\left[\mathcal{G}^{\text{ms-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b} \to 1\right] - \frac{1}{2}\right| \in [0,1]$.

Figure 16: Multi-show tracking game $\mathcal{G}^{\text{ms-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b}$ for a challenge bit b, and a winning condition win $\in$ {policy, attributes, consent}. Differences with the distinguishability game are shown in dark red.

- learn = g-leak corresponds to the adversary learning some function g of the attributes, and the advantage is the increase in the adversary's expected value of g over the baseline where it knows only the distribution of attributes. Here, g can be any function, e.g., g may specify a subset of attributes, a predicate over attributes, an interval or cluster, or give the adversary partial credit for approximate guesses. This corresponds to the notion of g-leakage in the QIF literature [5].

Lemma F.5 (Unbounded requests lead to unbounded inferences). *For any parameter κ, there exists an adversary $\mathcal{A}_{\mathsf{inf,full}}$ making $q = 1$ queries that wins the inference game $\mathcal{G}^{\mathsf{inf,full}}$ for full inference with advantage* 1.

Proof sketch. $\mathcal{A}_{\mathsf{inf,full}}$ picks a distribution over attribute vectors with min-entropy at least κ. During the challenge issuance step, it specifies that the challenge user is issued a credential with attributes drawn from this distribution. During the challenge presentation step, $\mathcal{A}_{\mathsf{inf,full}}$ requests all attributes, and sets B_{reg} and B_{pol} to be the identity function. Thus, the user proceeds and discloses

Setup.
(1) $\mathcal{A}$ selects a corrupted relying party $\tilde{\mathcal{R}}$ and a corrupted issuer $\tilde{\mathcal{I}}$.
(2) C creates an issuer $\mathcal{I}$ and users $\mathcal{U}_0, \mathcal{U}_1$, and initializes an instance of $\mathcal{F}_{\mathsf{EUDIF}}$ with issuers $\{\tilde{\mathcal{I}}\}$, users $\{\mathcal{U}_0, \mathcal{U}_1\}$ and RPs $\{\tilde{\mathcal{R}}\}$.
Challenge issuance. $\mathcal{A}$ instructs C to issue a credential to challenge users $\mathcal{U}_0$ and $\mathcal{U}_1$. $\mathcal{A}$ sends inputs $(\mathtt{ChalIss}, \mathsf{iid}, (\mathsf{attr_descr}_i)_{i\in[n]}, (\mathsf{attr}_i^{(0)})_{i\in[n]}, (\mathsf{attr}_i^{(1)})_{i\in[n]}, \mathsf{policy}_0, \mathsf{policy}_1)$ to C. The challenger sends two issuance requests to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of users $\mathcal{U}_0$ and $\mathcal{U}_1$ for the issuer $\tilde{\mathcal{I}}$ and the attribute types $(\mathsf{attr_descr}_i)_{i\in[n]}$. For user $\mathcal{U}_j$, $j \in \{0,1\}$, C programs the issuance oracle to return the attributes $(\mathsf{attr}_i^{(j)})_{i\in[n]}$ and the disclosure policy policy_j.
Challenge presentation. $\mathcal{A}$ may instruct C to request – for one of the two challenge users – access for an adversary-specified service for an adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, \mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]}, B_{\mathsf{reg}}^{(0)}, B_{\mathsf{reg}}^{(1)}, B_{\mathsf{pol}}^{(0)}, B_{\mathsf{pol}}^{(1)})$ to C. Challenger C then interacts with $\mathcal{F}_{\mathsf{EUDIF}}$ as follows. It sends a presentation request $(\mathsf{Req}, \mathsf{rid}, \mathsf{service})$ to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}_b$. In response to the $(\mathsf{Request}, \mathsf{service})$ message for the RP, it responds with $(\mathsf{aux} = (\mathsf{rp_info}, \mathsf{int_use}), f, (\mathsf{attr_descr}_i)_{i\in[n]})$. On input $(\mathsf{rp_info}, \mathsf{int_use}, f, (\mathsf{attr_descr}_i)_{i\in[n]})$ it sends $b_{\mathsf{check}} = 1$ (w.l.o.g, we assume users always check). On input $b_{\mathsf{registration}}$, the challenger returns $B_{\mathsf{reg}}^{(b)}(b_{\mathsf{registration}})$. On input b_{policy}, the challenger returns $B_{\mathsf{pol}}^{(b)}(b_{\mathsf{policy}})$. Finally, the challenger returns the response v (which may equal $\perp$) to the adversary.
Modeling strategies. We capture three winning strategies:
- win = policy: enforce $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$, and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$, otherwise the game returns 0.
- win = attributes: enforce $\mathsf{policy}_0 = \mathsf{policy}_1$, $B_{\mathsf{reg}}^{(0)} = B_{\mathsf{reg}}^{(1)}$, and $B_{\mathsf{pol}}^{(0)} = B_{\mathsf{pol}}^{(1)}$, otherwise the game returns 0.
- win = consent: enforce $\mathsf{policy}_0 = \mathsf{policy}_1$ and $f((\mathsf{attr}_i^{(0)})_{i\in[n]}) = f((\mathsf{attr}_i^{(1)})_{i\in[n]})$, otherwise the game returns 0.

Winning. $\mathcal{A}$ outputs a bit b'. The game outputs 1 iff $b = b'$.
Guess. $\mathcal{A}$ outputs a bit b'. The game outputs 1 iff $b = b'$.
$\mathcal{A}$'s advantage in the game is $\mathsf{Adv}^{\text{i-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}}} := 2 \cdot \left|\Pr\left[\mathcal{G}^{\text{i-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b} \to 1\right] - \frac{1}{2}\right| \in [0,1]$.

Figure 17: Issuer tracking game $\mathcal{G}^{\text{i-track,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}},b}$ for a challenge bit b. Differences with the distinguishing game are highlighted in dark red.

Setup.
(1) $\mathcal{A}$ selects a set of corrupted relying parties $\tilde{R}$, a tuple of attributes $(\mathsf{attr_descr}_i)_{i\in[n]}$, and a distribution $\mathcal{D}$ over attribute vectors for these attribute types, with the restriction that $H_\infty(\mathcal{D}) \geq \kappa$.
(2) C creates a "challenge" user $\mathcal{U}^*$ and an issuer $\mathcal{I}$, samples a tuple of attributes $(\mathsf{attr}_i^*)_{i\in[n]} \leftarrow \mathcal{D}$, and initializes an instance of $\mathcal{F}_{\mathsf{EUDIF}}$ with users $\{\mathcal{U}^*\}$, relying parties $\tilde{R}$ and issuers $\{\mathcal{I}\}$.
Issuances. $\mathcal{A}$ may instruct the challenge user $\mathcal{U}^*$ to request issuance of credentials from the issuer $\mathcal{I}$: $\mathcal{A}$ sends the message $(\mathtt{ChalIss}, (\mathsf{attr_descr}_i)_{i\in[k]}, \mathsf{policy})$ to C. C then sends an issuance request to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}^*$ for the issuer $\mathcal{I}$ and the attribute types $(\mathsf{attr_descr}_i)_{i\in[k]}$, and programs the issuance oracle to return $(\mathsf{attr}_i^*)_{i\in[k]}$ and policy.
Presentations. $\mathcal{A}$ may instruct the challenge user $\mathcal{U}^*$ to request access for an adversary-specified service for an adversary-controlled relying party $\tilde{\mathcal{R}}$, with an adversary-specified consent strategy. $\mathcal{A}$ sends the message $(\mathtt{ChalPres}, \mathsf{rid}, \mathsf{service}, B_{\mathsf{reg}}, B_{\mathsf{pol.}})$ to C. C then sends a presentation request to $\mathcal{F}_{\mathsf{EUDIF}}$ on behalf of $\mathcal{U}^*$ for the relying party rid and the service service, using the functions B_{reg} and B_{pol} to select the bits b_{reg} and b_{pol} that $\mathcal{U}^*$ sends to $\mathcal{F}_{\mathsf{EUDIF}}$ as part of the presentation interaction. $\mathcal{A}$ can adaptively repeat this step up to q times.
Guess. $\mathcal{A}$ outputs a tuple of attributes $(\widehat{\mathsf{attr}_i})_{i\in[n]}$.
$\mathcal{A}$'s advantage $\mathsf{Adv}^{\text{inf,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}}}(\kappa, q)$ in the game is defined as follows:
- If win = full: $\Pr[(\widehat{\mathsf{attr}_i})_{i\in[n]} = (\mathsf{attr}_i^*)_{i\in[n]}] - \max_{(\mathsf{attr}_i)_{i\in[n]}} \Pr[(\mathsf{attr}_i)_{i\in[n]} = (\mathsf{attr}_i^*)_{i\in[n]}]$
- If win = sensitive$((\overline{\mathsf{attr_descr}_i})_{i\in[\ell]})$: $\Pr[(\widehat{\mathsf{attr}_i})_{i\in[n]} = (\mathsf{attr}_i^*)_{i\in S}] - \max_{(\mathsf{attr}_i)_{i\in S}} \Pr[(\mathsf{attr}_i)_{i\in S} = (\mathsf{attr}_i^*)_{i\in S}]$, where S is the set of indices corresponding to the attribute descriptions $(\overline{\mathsf{attr_descr}_i})_{i\in[\ell]}$ classified as sensitive;
- If win = g-leak: $\mathbb{E}[g((\widehat{\mathsf{attr}_i})_{i\in[n]}, (\mathsf{attr}_i^*)_{i\in[n]})] - \max_{(\mathsf{attr}_i)_{i\in[n]}} \mathbb{E}[g((\mathsf{attr}_i)_{i\in[n]}, (\mathsf{attr}_i^*)_{i\in[n]})]$.

Figure 18: Inference game $\mathcal{G}^{\text{inf,win}}_{\mathcal{A},\mathcal{F}_{\mathsf{EUDIF}}}$ for an adversary $\mathcal{A}$, security parameter κ and query bound q.

all attributes to the adversary. The adversary thus learns the full attribute vector, and wins the game with advantage 1. □

Adaptivity. The above attack strategy is fully valid in the model of the functionality, where the adversary can adaptively query any attributes. We show that even in more constrained models (where the EUDIF functionality is restricted), similar attack strategies can be successful, albeit with a reduced advantage.

Mandating "not too revealing" presentations. Suppose that the EUDIF functionality requires that RPs must register only non-trivial functions of attributes (e.g., not of the form "disclose all attributes"). Further, suppose that issuers also specify policies that output 0 for any query of this type. In this case, an adversary very similar to the one above can still win the inference game with advantage 1 and a single query: this adversary registers a non-trivial function, and then still requests a presentation for the full attribute vector. Since nothing enforces that the user only discloses the output of the registered function, nor that the issuer policy must be respected, it is valid for the user to disclose the full attribute vector to the adversary, and the adversary wins with advantage 1. Beyond this obvious flaw, it is also not clear how to even define which functions are "too revealing" in general (and should be forbidden).

Enforcing "not too revealing" presentations. Suppose that the EUDIF functionality additionally disallows users to disclose attributes for presentation requests that were not registered by the RP. In this case, a slight tweak to the adversary from above still wins the inference game with advantage 1, albeit with more queries: the adversary picks a distribution over attribute vectors with min-entropy at least κ (for simplicity, assume that this distribution is a uniform distribution over κ bits). Then, the adversary picks q presentation functions/attributes such that any tuple of q presentations can be used to reconstruct the full attribute vector (e.g., the adversary picks $q = \kappa$ functions, each revealing a bit attribute). The adversary then registers these q functions, and requests q presentations for these functions. Since the user is allowed to disclose the output of the registered functions, it discloses these q outputs to the adversary. The adversary can then reconstruct the full attribute vector from these q outputs, and wins with advantage 1. This attack strategy can be made even more powerful by exploiting the fact that (i) the adversary can make its queries adaptively (i.e., it can use the output of one query to decide which function to query next), (ii) the adversary can corrupt multiple RPs, and thus register different functions at different RPs, (iii) the adversary can register new presentation requests at any time (e.g., also adaptively while making its queries).

G Updated Harm Tree

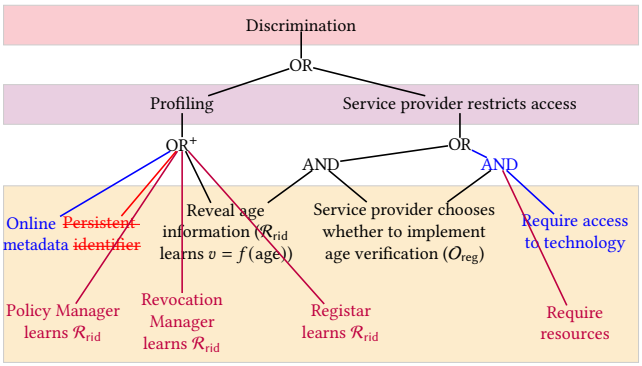


Figure 19: Age verification – Discrimination – Changes with respect to Fig. 6a after deployment, incl. the integration of PETs. The red component *persistent identifier* can be removed by the use of a PET that provides unlinkability (e.g., pseudonyms). There are four new dark red components. Three of them consist in new leakage: the policy manager, the revocation manager, and a registrar can learn the service the user wants to access because of new features introduced by deployment requirements. These features are: first a feature which allows the user to access the service's disclosure policy, secondly, the ability to revoke credentials, and thirdly the ability for the user to verify that the attributes requested by the service aligns with its registered intended use. These components and their leakage can lead to profiling and discrimination. The fourth new dark red leaf *require resources* comes from deployment requirements on the user's device, i.e., certain characteristics are now necessary to operate a wallet (e.g., secure element for device binding, compute power for advanced PETs) and potentially restrict access of some users, leading to discrimination.