How trustworthy is the Trusted Digital Identity Framework?

Evaluating security and privacy in Australian digital identity

Benjamin Frengley (1050642)

Supervised by Vanessa Teague

Submitted in partial fulfilment of the requirements of the degree of Master of Science (Computer Science)

75-point Research Project (COMP90070)
School of Computing and Information Systems
University of Melbourne

26 November 2020

Abstract

The Trusted Digital Identity Framework (TDIF) is a digital identity management system recently released by the Australian government. It features a brokered model in which authentication interactions happen through an intermediary known as an identity exchange, which allows the identities of the service provider and identity provider to be hidden from each other. This system is being integrated with government and private sector services, but no public analysis of its security and privacy properties yet exists.

In this work I offer an exploratory analysis of the TDIF, identifying a number of issues in its design and implementation and finding that the brokered model offers limited benefit at the cost of significant privacy issues. I present an concrete attack which allows an attacker to undetectably log in as a victim at a service provider of the attacker's choosing, and an attack on the identifiers used to recognise user accounts belonging to the same individual that allows an attacker to derive identity data about the user (such as a passport number) from the identifier. I also find many deviations between the specification and implementation, including violations of fundamental TDIF properties. I conclude with a strong recommendation that the Australian government considers alternative approaches to authentication such as a public key infrastructure-based system.

I certify that

- this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School
- the thesis is 24,870 words in length (excluding text in images, tables, bibliographies and appendices).

Acknowledgements

I acknowledge the custodians of the land on which I lived and studied, the Wurundjeri people of the Kulin Nation. I pay my respects to their elders past, present, and emerging.

Thanks to my supervisor, Vanessa Teague, for her endless patience, wisdom, and reassurances of "no, you're correct, that *doesn't* make sense"—without your kindness and guidance this work would not exist.

Thanks to my parents, Shona and Don, and my partner, Chika, for their love, support, and encouragement.

Thanks to Alex for being my sounding board even when nothing I was saying made sense, and to Meg for her relentlessly optimistic cheerleading.

List of Tables

List of Figures

Chapter 1

Introduction

To access most government services, Alice must be able to prove her identity. Before the Internet was widespread, this was solved relatively easily—Alice would take her government-issued identification documents to a government-approved location, such as a post office, and the staff there would verify that Alice matched the details included in the document and their records. However, with the rise of online services combined with the inherent anonymity offered by the Internet, an alternative solution is required.

Consideration of the requirements for proving identity online to the level required for government use began around the turn of the millennium [7], with solutions such as the Estonian identity card being rolled out not long after. Since then, many governments have implemented a variety of solutions to the problem, from the well-known system of electronic identity (eID) cards used in various European countries [2, 17, 34] to various solutions based on open standards such as SAML 2.0 [6], like New Zealand's RealMe[1]. The problem of proving your identity online—known as *digital identity management*—is not a static one, and so solutions must evolve over time—for example, four different versions of the aforementioned Estonian eID card have been issued since 2002 to address changing requirements and new use-cases [31].

The situation is, of course, no different in Australia. While business owners have previously been able to use AUSkey, a similar system which identified a user as being able to act on behalf of a business, there hasn't been a digital identity management system for general users to prove their identity to government services. This lack was discussed in the 2014 Financial System Inquiry [30], which identified a need for a cohesive digital identity infrastructure for both the public and private sector, rather than the fragmented network of credentials which has grown organically as services move online and policies and regulations for identity management and personal data handling evolve. The FSI recommended the development of a federated system, in which public and private sector entities provide identity verification for government and non-government services alike. To bring this recommendation to fruition, in 2015 the Australian government

[1] https://www.realme.govt.nz

created the Digital Transformation Office, now known as the Digital Transformation Agency (DTA). Under the digital identity umbrella program known as GovPass, the DTA began development of a federated digital identity management ecosystem, releasing the design as the Trusted Digital Identity Framework.

1 The Trusted Digital Identity Framework

The Trusted Digital Identity Framework (TDIF) consists of more than 500 pages of specifications across 13 documents[2], which together lay out a design for digital identity management shared between Australia's public and private sectors such that a single set of credentials can be shared across multiple services. It provides verification of identity documents (such as passports, birth certificates, and visas, among others) and other identity data about users, known as *attributes*, along with authentication of user identity. Following the recommendations laid out in the FSI, the TDIF uses a federated architecture in which a user can choose from one of a number of different identity providers to store and verify their identity data. While further identity providers will be potentially added in the future, only two currently exist: Australia Post's Digital iD[3], which can be used to access a variety of Australia Post and private sector services; and the Australian Tax Office's myGovID[4], which replaces AUSkey for accessing government services on behalf of a business.

One of the guiding principles of the TDIF is that it should be privacy enhancing [02]. It seeks to uncouple the identity of the user at the service and the identity provider to prevent either of those parties from tracking the user's behaviour across multiple authentication processes. The key mechanism for this property, which the TDIF refers to as *double-blind*, is the inclusion of an additional entity between the service and identity provider called the *identity exchange* in what is known as a brokered trust model [3]. Where in a typical federated system, the user would choose from one of several identity providers to use when they begin the authentication process at the service provider (in TDIF terms, the *relying party*), in the TDIF's brokered architecture the relying party instead sends the user to the identity exchange, which acts as an identity broker by conveying messages between the two ends. The identity exchange allows the user to select the identity provider they want to use; it then hands the user on while masking the identity of the originating relying party. The response messages from the identity provider also go through the identity exchange which then hides the identity provider and provides its own assertion of the user's identity. In this way, the two ends of an authentication process never learn about the other, thus preventing them from tracking the user's activity across multiple interactions.

The trust relationships within such a brokered federated model differ from other common systems. In a public key-based system such as those common in

[2] https://www.dta.gov.au/our-projects/digital-identity/trusted-digital-identity-framework/framework-documents

[3] https://www.digitalid.com/

[4] https://www.mygovid.gov.au/

Europe, a single central entity—the certification authority (CA)—issues keys to individuals which it then certifies. Each relying party trusts the CA, and can use the CA's public key to verify a user's key without requiring the CA's involvement at the time of the authentication process. Because the only party directly involved in the authentication process is the relying party, the same issue of user activity tracking does not arise. In a direct, pairwise trust model such as standard Single Sign-On (SSO)[5], the relying party has a direct relationship with each identity provider that it trusts, and thus it trusts an identity assertion received directly from that identity provider[6]. Because it must receive the assertion directly from the identity provider, the identity provider must be present in every authentication process, unlike in public key infrastructure. However, because the TDIF's brokered model does not allow a relying party to know the identity of the identity provider, a relying party cannot rely on that relationship to verify the authenticity of an identity assertion. Instead, the relying party must trust the identity exchange, which in turn trusts each identity provider with which it is integrated. An identity assertion from an identity provider is therefore trusted by the identity exchange, and since the identity exchange re-wraps the original valid assertion as its own, the relying party trusts the assertion it receives from the identity exchange without knowing the identity provider who originally provided it.

It is this brokered trust model along with the TDIF's recent release and Australia's tumultuous history of identity management [20] which makes the TDIF of particular interest. While direct, pairwise trust models and PKI are both relatively common and well understood, complex brokered identity systems like the TDIF are much less so. Despite the claim of the TDIF that it is privacy enhancing and secure, it shares many design similarities with other brokered models such as those used in the US (Federal Cloud Credential Exchange) and the UK (GOV.UK Verify) which have been found to have significant privacy and security flaws [4]. While the TDIF has been opened for industry and public feedback at various stages in its development, only summaries of the resulting changes have been released [10, 11], not the feedback itself. As a result, the TDIF has never undergone a publicly available analysis of its security. In a time where online privacy is increasingly entering public awareness, users want to be sure their data is being handled securely—and even more so when it involves sensitive identity data and important financial and government services. It is therefore clearly important to investigate the TDIF's claims of security and privacy to ensure that it provides adequate protections in such a sensitive space—this thesis aims to provide the first steps towards that end.

2 Contributions of this work

This thesis is the first published exploratory analysis of the Trusted Digital Identity Framework specification and ecosystem. As an exploratory work it

[5] Using, for example, OpenID Connect or SAML.

[6] Or received indirectly, such as through the user's browser, using mechanisms such as signing.

covers the TDIF widely, aiming to establish baseline understanding of the TDIF and its strengths and weakness rather than drilling into a single area in-depth; as such, it contextualises the TDIF among other, previously established government digital identity systems.

This work also presents three major novel contributions:

1. It presents an attack on the TDIF's deduplication process which allows an attacker to derive personal data such as passport numbers for any user for whom deduplication has been attempted.
2. It identifies a number of minor and two major deviations between the TDIF protocol and its implementations by myGovID and Digital iD, which call into question the integrity of the TDIF's accreditation process.
3. It presents an attack on the myGovID login process which allows an attacker controlling a malicious RP to log in at an arbitrary honest RP under the identity of a user who tries to log in at the malicious RP, which is aided by both the TDIF's double-blind design and the myGovID mobile app.

Finally, this work presents a number of smaller issues in the way the TDIF specification is written which make an already large and complex specification more difficult to comprehend for potential applicants.

3 Outline

This thesis is comprised of 9 chapters, the first of which is this introduction.

In Chapter 2 of this thesis, I provide a brief literature review of digital identity and its use in existing government systems.

In Chapter 3, I provide an overview of the TDIF, including its design, its architecture, the documents comprising it, and the roles it defines.

In Chapter 4, I discuss the role of the identity exchange and the privacy issues that arise as a result of a singular central entity being involved in many interactions.

In Chapter 5, I discuss the deduplication process defined by the TDIF to identify different identities which are owned by the same person, present an attack on the identifiers it uses for this process, and present a simple alternative approach.

In Chapter 6, I outline the OpenID Connect-based protocol used by the TDIF and discuss deviations from the protocol observed in the implementations by myGovID and Digital iD.

In Chapter 7, I present a code proxying attack against myGovID's login process which allows a malicious RP to undetectably log in as a victim at a different, honest RP of the attacker's choosing.

In Chapter 8, I identify a number of smaller issues in the writing of the TDIF specification and its implementations which obstruct the clarity of the specification for potential applicants.

In Chapter 9, I conclude the thesis by discussing the cumulative impact of the issues identified in the preceding chapters and recommending major changes to the TDIF's design.

Appendix A follows the bibliography, and describes a speculative but unverified attack on a naive RP integrated with myGovID which takes advantage of the TDIF's conflicting ID token validation requirements.

Chapter 2

Digital identity in literature

Digital identity has featured in literature for almost two decades.

The Liberty Trust Models Guidelines [3] were published in 2003, which concretely defined many of the common trust models found in digital identity systems today. Of particular interest is the *brokered/direct model*, which closely resembles the trust model used by the TDIF: in this model, a local service provider receives an authentication assertion from a remote identity provider with which it has not established trust, but it is able to trust the authentication assertion based on its trust in an intermediary which itself trusts the remote identity provider.

Many different identity management frameworks have come and gone, such as Identity Federation Framework (ID-FF), WS-Federation, BBAE, idemix, OpenID, and Microsoft Passport [5, 32, 42], while some prominent protocols used for federated identity management have lasted the distance, such as SAML [6].

Identity management is not an easy task. Dhamija and Dusseault [9] identified seven key challenges involved in identity management from the perspectives of both security and usability. These challenges are still relevant today and apply to the TDIF. For example, the TDIF includes a comprehensive concept of consent, but as the authors correctly identify, giving more fine-grained control over consent to users does not necessarily provide better control over their data as they become overwhelmed and numbed to the constant requests for consent. While the TDIF doesn't seem to avoid this issue, it does provide the ability to revoke consent at a future date, which perhaps helps mitigate this issue[7]. Maler and Reed [28] identify interoperability between different identity management standards as an issue, giving the example of SAML and OpenID. The TDIF explicitly includes this in its design by deliberately supporting both SAML and OpenID Connect; it uses the identity exchange as a place to simplify this interoperability by performing translation between the different standards, which allows a relying party supporting one standard to interact with an identity provider supporting the other at the cost of complexity at the identity exchange.

[7] Although it's arguable that it actually makes it worse, since it's just more consent complexity for users to deal with.

Digital identity in government

While digital identity management is very prominent in the private sector, it is similarly significant in the public sector, with governments worldwide opting to allow online access to their services and thus having to provide some means of authentication of user identity.

The prevalent means of digital identity management in Europe is the public key infrastructure-based electronic identity card, also known as eID cards, where a majority of countries in the European Union have eID card schemes. Arora [2] identify a number of the more interesting examples of eID schemes in the EU, such as the Austrian scheme in which multiple different types of cards (such as the Bürgerkarte and bank-issued ATM cards) are usable as eID cards [1]; the Belgian BELPIC, which uses three different X.509 certificates to achieve citizen authentication, electronic signatures, and card authentication; and the UK, which doesn't have an eID card scheme at all. The authors also note a number of issues which were unresolved at the time their paper was published, the key one of which is cross-scheme compatibility. This issue has since been addressed to a large degree by the development of the EU's electronic Identification Authentication and trust Services (eIDAS) regulation [34], which aims to provide compatibility between the eID schemes of participating countries.

Poller et al. [33] describe the German eID card, neuer Personalausweis, which is a three-function card similar to Belgium's BELPIC. The German card has a number of privacy-enhancing features, such as mutual authentication, where it requires clients to authenticate themselves before the card will release identity attributes and thus allows it to ensure that only the information relevant to that client is release; on-card verification, which releases identity attributes masked by predicates, such as age verification where only a yes or no is returned rather than an exact age; and restricted identification, which provides service-specific pseudonyms to prevent cross-service identity linking.

The Estonian eID card is one with a long history, first being rolled out in 2002 and being used by 67% of the Estonian population in 2018. However, any digital system with such a long history is bound to have security issues at some point—and, indeed, Parsovs [31] identified flaws in the Estonian card manufacturing process which resulted in different cards having the same private key, allowing the cardholders to impersonate each other. The authors also found situations in which cards had keys generated outside the card, which violates the Estonian card manufacturing regulations, and discovered a small number of cards which had corrupted 2048-bit RSA keys such that the keys were divisible by small prime factors.

PKI-based eID card schemes are not universal, with countries like New Zealand using a centralised SAML-based scheme [29], and countries like the United States and the United Kingdom using brokered models similar to the TDIF.

The brokered models in the US and UK are of particular relevance to the TDIF, as they are very similar architectures. As a result, the work of Brandão et al. [4] is very influential on this thesis. The authors evaluate the security and privacy properties of the hub-based brokered identity federations of the US (Federal Cloud Credential Exchange (FCCX), later renamed to Connect.gov and later renamed again to the current name, Login.gov) and the UK (GOV.UK Verify). The authors infer several desired security and privacy properties from the designs of the two systems and also identify a number of properties desirable in a "privacy-preserving" brokered federation in general; in particular, they identify several different aspects of unlinkability, such as edge unlinkability within a transaction (i.e., in any given transaction, the RP should not know the identity of the IdP and vice versa). They identify how each system fails to achieve many of these properties, sometimes requiring the meddling of a malicious hub but primarily by the design of the systems themselves, even when all parties are honest. Finally, they offer potential solutions which would allow both systems to achieve many of these properties, resulting in greater privacy for the users of these services even in the case of malicious hubs.

While there are differences in their design, the systems discussed in this work[8] strongly resemble the TDIF in many ways, and so much of the analysis of these systems also applies to the TDIF. It's clear that the TDIF also fails to achieve many of these properties, both in theory and in existing implementations, which is discussed in detail in chapter 4. Many of the solutions offered in Brandão et al.'s work are equally applicable to the TDIF.

The authors note a need for a comprehensive solution for brokered identification in the form of a formal specification with an unambiguous protocol and proven security. The TDIF having many of the exact same issues as the discussed systems (almost word-for-word the same as described by the authors two years before the TDIF was published!) strongly emphasises this point. However, as the authors frame the issues from the perspective of user privacy, it is worth considering *why* the same issues appear consistently across different designs—in particular, whether governments are likely to implement a protocol which prevents surveillance of user activity.

[8] In particular the FCCX, as neither it nor the TDIF have a matching service like GOV.UK Verify.

Chapter 3

The Trusted Digital Identity Framework specification

The Trusted Digital Identity Framework lays out, in more than 500 pages across 13 documents, the design for an Australian digital identity management framework which spans the public and private sectors. It establishes an *identity system*—an online ecosystem for managing digital identity in which all parties trust each other based on the assertions of some authoritative source [14]—as well as the *trust framework* which governs it—the set of rules and specifications to which they must all adhere [14, 27].

This chapter is divided into two main parts. In the first, I provide a brief summary of the development history, structure, and nomenclature of the TDIF specification documents. In the second, I give an overview of the identity ecosystem specified by the TDIF, its guiding principles, architecture, participant roles, and existing members of the TDIF which are available for public use today.

1 Framework documents

1.1 Release history

While the Financial System Inquiry recommendation to develop a digital identity strategy came in 2014, the first draft of a TDIF component came in mid-2016 when the initial overview document was released. A series of internal reviews and public feedback rounds followed over the next two years before the release of TDIF version 1.0 in February 2018[9]. Despite the 1.0 label, this version of the TDIF was still not complete, with the second part being released in September 2018[10] and a third part in April 2019[11]. The most recent update came in

[9] https://www.dta.gov.au/news/digital-id-another-step-closer
[10] https://www.dta.gov.au/news/trust-framework-milestone
[11] https://www.dta.gov.au/news/third-release-trusted-digital-identity-framework

May 2020[12], when the Digital Transformation Agency (DTA) released a major overhaul of the specification, compressing the 19 documents from the April 2019 release into 13 new documents with a different structure and a more formal way of representing requirements; several minor revisions have since followed.

At the time that this research began, the current version was the April 2019 release. Because the TDIF's naming for different versions of the specification is confusing and inconsistent, with no formally-indicated name for a release prior to the May 2020 update and different documents being labelled different versions depending on their original release date and time of latest update, the rest of this work will refer to the April 2019 release as TDIF v1.5, based on the version of the Overview document at that time [14]. The May 2020 release is explicitly labelled as "Trusted Digital Identity Framework Release 4", so this work will refer to it as TDIF Release 4. This work will focus on TDIF Release 4 because it is the current release. Because the changes are primarily around the way the specification is written rather than substantive content, this work does not discuss the content changes made between TDIF v1.5 and TDIF Release 4; however, chapter 8 discusses how the changes between releases impact the clarity of the specification.

Several times during the course of development, the draft specification at the time was opened for public consultation. These feedback rounds resulted in "more than 2,450 comments" [10] from a range of sources, which helped guide the development of new releases. However, while changes made during each round, presumably at least partially based on the feedback, have been released [10, 11], the feedback itself has not. This makes it very difficult to know what has been noticed or commented on, how thoroughly the TDIF has been investigated at each stage, or how much of the feedback has actually been used. This lack of transparency is a major driving force behind this work; we believe it imperative that a system covering such a widespread domain and handling personal data for many people should be subject to careful scrutiny in the public eye, which the feedback process of the TDIF development has not adequately achieved by keeping the feedback private.

1.2 Specification documents

TDIF Release 4 contains 13 documents grouped into three loosely-defined categories: governance, which includes documents dealing with accreditation; requirements, which lay out the formal requirements participants in the TDIF must meet; and guidance, which provide informal information to help participants meet the requirements.

Two of the TDIF documents provide high-level context of the TDIF specification: the Overview (a guidance document) [02] and the Glossary (a governance document) [01], which in TDIF v1.5 were the same document [14]. Three requirements documents provide the formal requirements which must be met by appli-

[12] https://www.dta.gov.au/news/fourth-release-trusted-digital-identity-framework

cants to the TDIF [04, 05, 06], each of which has a respective guidance document which provides informal context and information about the requirements [04A, 05A, 06A]. Two of the requirements documents provide TDIF-specific profiles for the protocols on which the TDIF builds, OpenID Connect [06B] and SAML [06C], while a governance document provides a profile of the user attributes handled by the TDIF [06D]. The final two documents describe the accreditation process applicants must go through to join the TDIF [03, 07], one of which is a governance document and the other of which is a requirements document.

1.3 Requirements

Previous to TDIF Release 4, requirements were laid out without an obvious delimiter between different requirements. A requirement was distinguished based on its use of certain specific keywords, the meaning and format of which is laid out clearly in a dedicated section of the Overview. Because the keywords can also be used outside the context of a requirement, use in a requirement was distinguished by using all uppercase letters in both bold and underlined style.

> Specific conventions are used across the TDIF documents to signify requirements. The following section defines these words and how they should be interpreted. TDIF documents that include these conventions will include this phrase at the beginning of the document under the title "Conventions":
>
> The key words "**MUST**", "**MUST NOT**", "**SHOULD**", "**SHOULD NOT**", and "**MAY**" in this document are to be interpreted as described in the current version of the TDIF: Overview and Glossary. (*TDIF Overview and Glossary*, §4.7)

TDIF Release 4 restructured requirements to make them more clearly distinct and able to be individually referenced by providing every requirement with a unique identifier indicating the subject area, document section, requirement, and sub-requirement [02, §2.11]. For example, a requirement named PRIV-03-04-01a would refer to the privacy (PRIV) requirement 01a in section 3.4 (03-04) of the relevant document. The new format also includes the month and year in which the requirement was most recently updated and the participants to whom it applies.

> The following is an example of a TDIF requirement:
>
> **TDIF Req**: PRIV-03-04-01a; **Updated**: Mar-20; **Applicability**: A, C, I, X
>
> An Applicant, not covered by the Privacy Act, *MUST* report eligible data breaches as defined in the Privacy Act 1988 to affected Individuals and the Oversight Authority and DTA. (*02 - Overview*, §2.11)

The keywords used in TDIF convention were also changed in TDIF Release 4. The most important of these changes is the complete removal of "**SHOULD**" and "**SHOULD NOT**", the consequences of which are discussed further in chapter 8. The dedicated section describing the meaning of these keywords was also removed in favour of placing their definitions in the general glossary [01]. The format also changed from bold keywords to italicised: where a keyword would have been "**MUST**", the current style is "*MUST*".

2 The TDIF ecosystem

2.1 Guiding principles of the TDIF

The TDIF is designed and operates according to eight guiding principles [02, §2.3]:

- User centric
- Voluntary and transparent
- Service delivery focused
- Privacy enhancing
- Collaborative
- Interoperable
- Adaptable
- Secure and resilient

The extent to which these principles are adhered to in practice varies. The *voluntary and transparent* principle dictates that users choose whether or not to participate in the TDIF ecosystem; however, the reality is that if a user wishes to access a number of government services on behalf of a business they have no choice but to use myGovID for authentication, forcing them to participate. Similarly, because the sets of services which can be accessed using existing identity providers is entirely disjoint, a user participating in the TDIF never has a choice of which identity provider to use, which breaks the *user centric* principle. When using services which use myGovID for authentication, users have no ability to manage consent, a key component of the *privacy enhancing* principle. It's unclear what the justification behind violating its own principles is or whether it will adhere more closely in the future as the ecosystem develops and expands.

2.2 Architecture

The architecture established in the TDIF is an identity federation, in which users have a choice from multiple competing identity providers from both the public and private sectors, allowing them to select who they trust with their data. Unlike most identity federations, but like the US' FCCX and UK's GOV.UK Verify, it uses a brokered model. In a normal identity federation, a service provider integrates directly with identity providers, referred to in the TDIF as being "one-legged". In a brokered model, however, service providers and identity providers do not integrate with each other. Instead, all authentication

interactions are mediated by a "broker", a third party which sits between the service provider and identity providers and keeps them separate from each other. In contrast to the "one-legged" style of direct integration, the TDIF refers to this as "two-legged": the first "leg" is the integration between the service provider and the broker, and the second leg is the integration between the broker and the identity provider.

The TDIF uses this brokered model for several reasons. The first is simplicity: a broker integrates with multiple identity providers, so a service provider that integrates with a single broker effectively integrates with every identity provider connected to that broker. This makes joining the TDIF much simpler for a service provider; since there are relatively many service providers compared to the number of identity providers and brokers, making it easy for them to join the TDIF federation is important. An additional simplicity benefit is that the broker provides a single centralised point at which user consent to release personal data to the service provider can be gathered, rather than requiring each identity provider implement it themselves[13].

The primary purpose of using a brokered model, however, is to separate the identity provider and service provider to protect the privacy of users. When the service provider connects directly to identity providers, the identity providers are capable of tracking user activity across multiple different service providers[14]. By placing a broker between them, it can act as a privacy barrier by masking the identity of each end of the interaction and therefore making it difficult for an identity provider to determine which service is being used. The TDIF refers to this property as *double-blindness*.

As can be seen in figure 3.1, the role of the identity broker is fulfilled by an *identity exchange*, which mediates between the *relying party* which provides services, and the *identity service provider*. The identity provider itself integrates with a *credential service provider* which handles user credentials and binds them to an identity; the identity provider itself binds the identity from its credential service provider to create a digital identity for a user [06A, §1.1]. It also integrates with one or more *attribute verification services* which verify identity data, referred to as *attributes*. The TDIF also includes an entity called an *attribute service provider* which integrates directly with an identity exchange and provides specialised attributes outside the scope of those handled by the identity service provider, such as qualifications or attributes belonging to non-person entities such as businesses.

A simplified version of the authentication process used by the TDIF is as follows:

1. A user accesses a service they wish to use at a relying party integrated with the TDIF, where the user begins the TDIF authentication process.

[13] However, this does mean consent isn't required to release data to the broker—or, if it *is* required, there's duplication between the two points.

[14] And vice versa. The opposite direction is less concerning because users are unlikely to use multiple identity providers with a single service, but using one identity provider for multiple services is common.

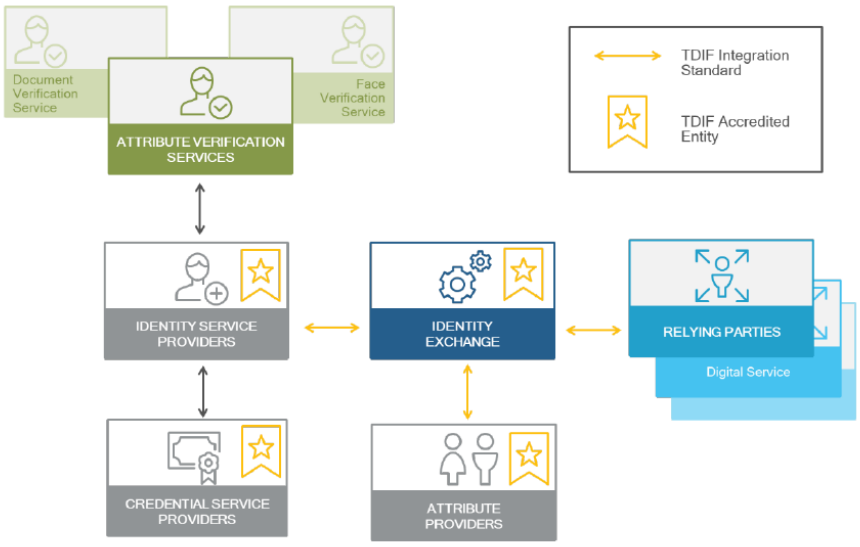

Figure 3.1: TDIF Identity Federation Conceptual Architecture [06A, §1.1]

2. The user is sent from the service to the identity exchange along with a list of requested attributes, where they are offered a choice of which identity provider they wish to use to authenticate.
3. The user selects an identity provider and is sent on by the identity exchange along with the list of requested attributes but with the identity of the relying party masked.
4. At the identity provider the user authenticates themself, such as by entering their username and password.
   (a) If the user has not verified their identity using identity documents at this identity provider before, they may do this now; they can also do this outside an active authentication process.
5. The identity provider returns the user to the identity exchange along with an authentication assertion signed by the identity provider and the values of the requested attributes.
6. The identity provider displays the list of attributes to the user and requests their consent to release them to the relying party.
   (a) If the identity provider is unable to provide all of the requested attributes, the identity exchange may first request some from an attribute service provider as appropriate.
7. Once the user has consented, the identity exchange returns the user to the relying party along with the authentication assertion (re-signed by the identity exchange to mask the identity of the identity provider) and the values of the requested attributes.
8. The relying party accepts the authentication assertion and allows the user to access the service.

2.3 Accreditation

The TDIF also defines an extensive accreditation process which is designed to enforce that applicants for accredited roles appropriate implement their role's requirements. Accreditation is handled by the *Oversight Authority*, a role which is fulfilled by the DTA. The accreditation process consists of an initial accreditation which must be carried out before an applicant can join the TDIF [03], and ongoing accreditation obligations, which are annual assessments designed to ensure that accredited applicants continue to meet the expected requirements [07].

2.4 Roles in the TDIF

Within the TDIF there exist four accredited roles: Identity Service Providers, Identity Exchanges, Credential Service Providers, and Attribute Providers. Of these, this work focusses primarily on Identity Service Providers and Identity Exchanges, as they are the major roles which provide the main functionality. This work also focusses on the role of Relying Parties, as they are the consumers of the authentication functionality provided by the TDIF ecosystem.

Relying Parties

A *Relying Party* (in this work referred to primarily as an RP) is a user-accessible digital service (or related collection thereof) which integrates with the TDIF to gain access to its authentication and identity proofing capabilities. RPs are not included in the accreditation process of the TDIF, although the protocol they must implement to integrate with an Identity Exchange is specified.

Identity Service Providers

Identity Service Providers (IdPs) provide authentication of user identity and verification of their attributes. They manage user accounts and integrate with various attribute verification services to validate user identity documents and other identity data. IdPs integrate with a (or operate their own) credential service provider to bind a user account to a set of credentials which the user can use to authenticate themselves to the IdP; the IdP then provides the authentication assertions consumed by the rest of the TDIF ecosystem.

Identity Exchanges

Identity Exchanges (IdXs) sit in the centre of the TDIF ecosystem and mediate interactions between RPs, who consume authentication assertions, and IdPs, who provide them. An IdX fulfils a number of roles, including handling consent, enforcing the double-blind property, and auditing—these are discussed in detail in chapter 4.

Credential Service Providers

A *Credential Service Provider* (CSP) generates, stores, and distributes credentials which are used authenticate a user against their digital identity within the TDIF.

Attribute Service Providers

Attribute Service Providers (ASPs) are dedicated entities to handle user attributes, often providing specialised attributes that would not normally make sense for an IdP to store. They integrate with an IdX as if they were an RP so that they may leverage TDIF authentication, but they also provide a REST API via which the IdX can request attributes. ASPs can also be accessed directly by RPs rather than going through the IdX, although the IdX still needs to mediate the process of getting a security token which can be redeemed at the ASP by the RP.

2.5 Existing participants

At the time of writing, there exist four[15] accredited services within the TDIF: Digital iD and myGovID, both of which are IdPs and have an associated CSP, run by Australia Post and the Australian Tax Office respectively; an IdX ostensibly run by the Department of Human Services; and the Relationship Authorisation Manager, an ASP managed by the ATO.

Digital iD

Digital iD is an IdP developed and managed by Australia Post used for accessing private sector services. It is integrated with a wide range of RPs, from Travelex to the Australian Electoral Commission to Australia Post itself. Digital iD also provides a CSP in the form of a mobile app, through which users can sign up, verify identity documents, view their authentication history, and provide consent for authentication requests.

myGovID

myGovID is an IdP developed and managed by the Australian Tax Office used for accessing government services. It is integrated with a long list of government-run RPs[16], in most of which a user is acting on behalf of a business. myGovID provides a CSP as a mobile app through which users can sign up, verify identity documents, and accept authentication requests. In order to facilitate acting on behalf of businesses, myGovID integrates the with Relationship Authorisation Manager, which is an ASP providing business authorisation attributes.

[15] Or six, if you count the CSPs as separate entities.

[16] The full list can be seen at https://www.mygovid.gov.au/where-to-use-it.

Relationship Authorisation Manager

The Relationship Authorisation Manager (RAM) is an ATO-run government service which manages authorisations for individuals to act on behalf of businesses online. It acts as both an ASP and an RP in the TDIF, and is involved in almost all current uses of myGovID.

The identity exchange

There is currently one accredited identity exchange, ostensibly run by the Department of Human Services. Since it is not a directly user-accessible service[17], it is hard to verify much about it.

[17] Although it should be, as it is required to provide a dashboard through which users can view and manage historical consent.

Chapter 4

Privacy Considerations at the Identity Exchange

The identity exchange is arguably the key component of the TDIF's design. An IdX is involved in every interaction in the TDIF[18], and is responsible for maintaining the double-blind property that the TDIF aims to have. However, due to the IdX's participation in all authentication processes combined with its many roles covering a broad range of important functions, its presence is not without impact on privacy.

The TDIF is not the first occurrence of a brokered model in a government identity system: the United States government's *Federal Cloud Credential Exchange*[19] (FCCX) and the United Kingdom government's GOV.UK Verify both use similar models to the TDIF, although each is slightly different. A number of privacy and security failings s caused by the presence of a broker were identified in the FCCX and GOV.UK Verify by Brandão et al. [4], many of which apply in similar ways to the TDIF.

This chapter discusses some of the privacy implications that stem from the IdX's position as identity broker in the TDIF. First, I outline the roles the IdX fills in the TDIF. I then discuss the work of Brandão et al. and how it applies to the TDIF. Finally, I identify and discuss some privacy issues unique to the TDIF.

1 Role of the Identity Exchange

In a typical identity federation, an RP would integrate directly with one or more IdPs and the user would select which one to use based on their personal preference and the RP they're accessing. However, in the TDIF the RP instead integrates only with an IdX, which acts like an IdP to the RP. When the user goes to access the RP, it redirects them to the IdX where the process effectively

[18] With a few exceptions in very specific circumstances.

[19] The FCCX was a pilot program for an identity ecosystem. It was then implemented as a real system as Connect.gov, which has since been replaced by the current system, Login.gov.

begins again, this time with the IdX acting as a more typical OIDC RP integrated with one or more IdPs. The IdX allows a user to select their choice of IdP and redirects them accordingly, where they authenticate themselves to the IdP and are redirected back, as would occur in a standard OIDC authentication process. The IdX then redirects them back to the RP along with an ID token proving the user's authenticity, whereupon the user may continue to use the RP's service.

The complicated two-legged authentication process serves to provide several things, the key among which is the aforementioned double-blind property.

"Double blind" federation
: An IdX mediates all interactions between an RP and an IdP such that they should never know each others' identities. By requiring all interactions go through the IdX, it in theory guarantees that all IdPs will appear the same to the RP, thus preventing an RP from gathering information on the user's choices. The same occurs in the other direction: to the IdP, all authentication requests come from the IdX and the RP is unknown.

Identifier management
: To ensure that a user appears the same to an RP regardless of their choice of IdP, the IdX generates a consistent pseudonymous identifier for the user known as an *RP Link* and provides it to the RP, regardless of the IdP used. It is responsible for maintaining a mapping between the user's identity at the IdP (identified by an IdP-provided *IdP Link*) and at the RP (identified by the IdX-provided RP Link).

Deduplication
: In order to ensure the RP Link for a user is consistent across multiple IdPs, the IdX must determine when two user identities at different IdPs belong to the same user and map the IdP Links to the same RP Link. This process is referred to by the TDIF as *deduplication* and is discussed in detail in chapter 5.

Attribute visibility
: User attributes may be spread across multiple sources depending on the type of attributes. The IdX is responsible for gathering all the necessary attributes for an authentication process, and thus it is also responsible for informing a user of what attributes are being sent to the RP.

Computed attributes
: In some cases, an RP does not require attributes at the full level of detail, instead only needing to know that the user meets some criteria (e.g., they are older than some limit). The IdX is responsible for converting the original attribute value into the lower detail form, which is called a *computed attribute*.

Consent
: Since the IdX is the entity which gathers all of the attributes, it is also

responsible for gathering the user's informed consent to release these attributes to the RP. The IdX is also responsible for tracking historical consent, as some attributes may not require that consent is explicitly given each time and consent may be revoked by the user.

Auditing
The IdX stores information about authentication interactions to allow auditing and investigations of criminal activity. This includes when an interaction occurred, which attributes were shared, and which parties were involved.

Ease of integration
By only requiring that an RP integrate with a single IdX in order to be able to use all of its associated IdPs, the IdX simplifies the process of joining the TDIF federation.

1.1 Double-blindness—or an all-seeing eye?

Foremost among the roles described above is the enforcement of the TDIF's double-blind property. The IdX is the mechanism through which the identity of the IdP and RP are hidden from each other, and implicit in this is that the IdP and RP must trust the IdX—should the IdX be malicious, it corrupts not only the RP's relationship with the IdP, but the RP's relationships with every other IdP and the IdP's relationship with every other RP.

The system's deep trust in the IdX has privacy implications which spread through the design of the entire TDIF. The IdX is a participant in all authentication interactions, and it sees all information sent between an RP and an IdP, which gives it full view into all identity data. Even though the TDIF allows OIDC ID tokens and access tokens to be encrypted when returned from the IdP [06B, OIDC-03-06-11], they are encrypted with the IdX's public key to allow the IdX to decrypt them so that it may replace information which identifies the IdP with its own. There is no mechanism through which a token may be encrypted by the IdP for only the RP to decrypt, so there is no way to avoid the IdX having full access to all data that travels through it. Indeed, the TDIF is specifically designed such that the IdX *must* have access to all attributes which flow through it, as it is responsible for informing a user what attributes are being accessed and for generating computed attributes [06, FED-05-02-02]. The IdX is further required to record significant audit histories of user interactions and consent, including who was involved, when it occurred, and the attributes requested and returned [06, §2.1.3].

The TDIF refers extensively to the privacy advantages of its double-blind design, but it spends very little time considering the implications of having a singular central entity be party to all messages sent through the TDIF federation, their contents, and the identities of senders, receivers, and users. The IdX effectively exists outside the TDIF's model of privacy, and the trust in it is absolute. It's unclear exactly what entitles it to more trust than any other

TDIF-accredited entity, given that the only things preventing it from collecting information about user activity[20] are the regulations set out in the TDIF[21], rather than technical means which it could not circumvent.

The IdX is intended to place a "privacy barrier" between the IdPs and the RPs. Given the IdX's omniscience and omnipresence, however, this privacy barrier seems more like a mere relocation of the privacy issues. Instead of IdPs illicitly tracking user activity, the IdX's role explicitly requires it to view all user data and track user activity—by design! Because an IdX is designed to integrate with multiple IdPs, it has far more power to track activity more widely than an IdP alone. Due to its role of deduplicating user identities across different IdPs, the TDIF can even track a single user across multiple IdPs, which IdPs would not be able to do themselves without collusion.

It is clear that the presence of the IdX does not have the clear-cut privacy benefits that the TDIF claims. It replaces the concern of possible tracking by one IdP with explicitly designed tracking across multiple IdPs. This situation could be generously attributed to a mere lack of understanding on the behalf of the DTA, but given that researchers had already observed the same issues in other brokered designs developed by governments by the time the TDIF was under development, the repeated pattern of governments designing a "federated" system in which all information is nevertheless channelled through a government-controlled entity begins to raise the question of whether it is intentional.

2 Issues shared with the FCCX and GOV.UK Verify

The work of Brandão et al. [4] offers a very valuable starting point for considering the privacy implications of the identity exchange. Both the FCCX and GOK.UV Verify are architecturally very similar to the TDIF, and aim to achieve much the same goals. In both, like the TDIF, a broker mediates between an RP and an IdP in order to hide the identity of each party from the other; in the FCCX this property is called *unlinkability*. Also like in the TDIF, the FCCX and GOV.UK Verify do not comprehensively describe the privacy and security properties they aim to provide. The authors instead infer a number of desired properties based on the design of the systems, which apply much the same to the TDIF.

While the authors outline a number of potential solutions to the problems they describe, a comprehensive analysis of how these (or similar) solutions may be adapted to the TDIF is left as future work.

Authenticity

The RP should, on completion of the authentication process, be able to trust that it has a valid session with the user and that the attributes and pseudonymous identifier (the RP Link) are valid for that user. This property relies on the

[20] More than it is already required to collect, that is.
[21] Which could just as easily be applied to an IdP if there were no IdX in the picture.

IdX being trustworthy, since the RP must trust that IdX is returning attributes associated with a genuine digital identity at a trusted IdP.

The authors identify four ways in which a malicious hub can violate authenticity by impersonating a user, three of which apply to the TDIF. While the TDIF avoids their proposed "impersonation at intended RP" attack due to the use of a secret tied to the user agent (browser) session [06B, OIDC-02-06-02], it remains vulnerable to the other impersonation attacks they describe. Due to the IdX's full visibility into the user's attributes and pseudonym, a malicious hub can replay the information it has previously observed to impersonate the user at an arbitrary RP without requiring user authentication, as it can initiate a session at the RP and generate an undetectably-forged valid authentication assertion. This results from there being nothing tying the assertion returned from the IdP to the assertion returned from the IdX, which allows the IdX to generate an assertion without a matching IdP assertion; due to the RP's trust in the IdX, this goes undetected.

Edge unlinkability within a transaction

This property is directly analogous to the TDIF's double-blindness. The authors define edge unlinkability within a transaction as holding if: the IdP does not learn the identity of the RP; the RP does not learn the identity of the IdP; and the IdP and RP do not learn the pseudonymous identifiers for the user at the other party. Like in the FCCX and GOV.UK Verify, in the TDIF the IdX knows all of these for each transaction.

In theory, this property does hold for the TDIF. However, in practice it fails in a number of ways. First, in the TDIF ecosystem as it exists today, the IdX is integrated with only a single IdP (myGovID) and the RPs clearly indicate that a user will authenticate using that IdP, so it is clear that the RP has knowledge of the IdP[22]. In the case of the IdP which is not integrated with the IdX (Digital iD), the IdP and RP integrate directly so have clear knowledge of each other. In addition, even in myGovID where the IdX is present, the user pseudonym does not always differ between the IdP and the RP (i.e., the RP Link is the same as the IdP Link)[23], violating the third component of the edge unlinkability property.

Unlinkability by the hub

The authors identify this property as desirable, rather than inferred from the design of the FCCX and GOV.UK Verify. The general form of this property is that the hub should not be able to link the same user across transactions. The authors identify several different forms, as described below.

[22] See chapter 7.
[23] See section 51.

Weak unlinkability across RPs: The hub cannot link transactions of the same user across different RPs, where a user is identified by a digital identity at an IdP. This property is not satisfied by the TDIF (and nor is it satisfied by the FCCX or GOV.UK Verify), as the IdX is responsible for maintaining a mapping between IdP Link (which identifies the user at the IdP) and RP Link [06A, §2.2.1.1], and therefore by design links the user across different RPs.

Weak unlinkability across IdPs: The hub cannot link different transactions across different digital identities at different IdPs when the transactions resolve to the same user account at a given RP. The TDIF (as of TDIF Release 4) fails to satisfy this property by design, as the IdX is explicitly responsible for deduplicating different digital identities at different IdPs into a single RP Link. In chapter 5 I discuss this process in detail, and propose a tradeoff which allows users to opt in to this linking, similar to that which the authors note is present in the FCCX.

Strong unlinkability: The hub cannot link transactions where the same user account at the IdP is used to access the same user account at the RP. None of the FCCX, GOV.UK Verify, or the TDIF satisfy this; the aforementioned identity mapping at the IdX combined with its requirement to store a history of user interactions allows it to link this across transactions.

Edge unlinkability across transactions

The authors extend the inferred property of edge unlinkability within a single transaction to also apply across multiple transactions.

Across two transactions with the same user account at an IdP: The IdP should not be aware of whether the RP has changed across multiple transactions with the same user account. The authors note that this property can be derived from the design of the FCCX and GOV.UK Verify, and the same is true of the TDIF. In practice, however, myGovID leaks the identity of the RP to the IdP through the HTTP Referer header[24] while Digital iD does not use an IdX at all, and therefore the TDIF as currently implemented does not meet this property.

Across two transactions with the same user account at an RP: The RP should not be aware of whether the IdP has changed across multiple transactions with the same user account at the RP. The deduplication process implemented by the IdX should provide this property in theory, as different IdP Links will be resolved by the IdX as referring to the same user and therefore the same RP

[24] See chapter 7.

Link will be provided to the RP. However, the deduplication process frequently fails[25], so it is not effective at providing this property.

Across transactions with the same user account at an IdP but different RPs: colluding RPs should neither be able to identify a shared user based on their list of known user pseudonyms nor be able to predict a user's pseudonym at another RP even if they know that their respective pseudonyms correspond to the same user. The TDIF dictates that different RPs should receive different RP Links, which is facilitated by the IdX. It previously required that RP Links should not be derived in any way from IdP Links [12, §2.3.2.1], but since TDIF Release 4 has instead delegated to OpenID Connect pairwise identifiers [06, FED-02-03-01][39, §8.1], which do not have a concept of the separation between RP Link and IdP Link due to the direct relationship between RP and IdP in OpenID Connect. In practice, myGovID has been observed to issue identical RP and IdP Links, clearly violating this property.

Attribute privacy

The authors define this property in two parts: access to attributes should be limited to the minimum necessary to provide the required information, such as providing only age predicates rather than a date of birth; and that the hub should not have visibility into the attributes being exchanged or verified.

The first of these can be inferred from the TDIF, as it defines the concept of computed attributes [06D, §3.4]. However, the second is clearly not compatible with the TDIF's design as the IdX is one of the places at which raw attributes can be reduced to computed attributes. The IdX is also responsible for gathering user consent to release attributes and is required to record the attributes requested and returned in any given authentication process, so it must have access to attribute data. Nevertheless, it seems desirable that the IdX should not have or require access to this data due to the significant privacy impact that results from it having this access.

3 Issues unique to the TDIF

While the TDIF's brokered model shares obvious design similarities with other existing brokered models, it was designed specifically to fulfil the needs of the Australian identity management space and, as a result, it also has its own unique properties and considerations.

3.1 The presence of multiple identity exchanges

Unlike the FCCX and GOV.UK Verify, the TDIF is explicitly designed to support multiple, separate identity exchanges simultaneously. It is intended that differ-

[25] See chapter 5.

ent IdXs which service different domains, such as the financial sector or state governments, can exist and be operated by parties from that domain. These separate identity exchanges and the domain they service are known as *Identity Sectors* [12, §2.3.1.2][26].

With a combination of appropriate oversight and technical limitations on the ability of an IdX to monitor the details of the traffic and interactions it processes, this would be an entirely reasonable idea. However, the technical limitations are in many cases absent or explicitly omitted to allow other functionality. While there is an accreditation process which aims to provide oversight of TDIF applicants, it is flawed and many violations of requirements seemingly escape it in practice—a particularly clear example of this is Digital iD's complete omission of an IdX. Given the magnitude of violations which can make it through the accreditation process, it is concerning that the TDIF would consider giving almost unfettered surveillance power to the private sector.

[26] Interestingly, TDIF Release 4 contains no mention of the identity sector concept. However, given that it does still explicitly mention the presence of multiple identity exchanges [06A, §2.1.1] and there is no indication in the release changelog [10] of the removal of this concept, it seems likely to be an oversight.

Chapter 5

Identity deduplication using Evanescent Deterministic Identifiers

The TDIF defines a process called *deduplication* to associate accounts at different IdPs belonging to the same user using Evanescent Deterministic Identifiers (EDIs). However, the specification for this process has a number of major issues, the most egregious of which allows an attacker to learn highly sensitive data about any user for whom an EDI has been derived. These issues were reported to the DTA on 14 September 2020, but no clear response was received other than an acknowledgement that they had received the report.

In this chapter, I first outline the deduplication process and how EDIs are generated and used. I then identify and discuss two issues which significantly limit the circumstances in which deduplication is possible at all, and one security issue in the EDI generation process which allows an attacker with only moderate resources to derive sensitive data from an EDI. Finally, I raise the question of whether deduplication need be automatic at all and offer an alternative approach which fits better with the TDIF's guiding principles.

1 The deduplication process

The TDIF allows users to log into the same RP using different IdPs as sources of authentication. In order to ensure that the RP sees the same user (as represented by the RP Link [06A, §2.2.1] returned by the IdX) regardless of which IdP they used, the TDIF also defines a *deduplication* process [06A, §2.2.1.2], the purpose of which is to allow an IdX to determine when a user's identity at an IdP corresponds to an already encountered identity at a different IdP for the same RP.

The deduplication process revolves around an attribute called an Evanescent Deterministic Identifier (EDI). An EDI is an identifier for a user whose value is

derived from verified information about that user, which aims to allow different IdPs to independently arrive at the same EDI for the same user, without needing any information about the user beyond what the user has provided to each IdP.

The deduplication process first appeared in TDIF Release 4, and was accompanied by a rewording of the TDIF's guiding principles which weakened the principles allowing users to maintain separate digital identities. It's not clear if this is directly related, due to the TDIF's tendency to not release the reasons behind changes made between versions. However, given that the deduplication process deliberately removes the ability of users to choose when their digital identities are merged, it does not seem unreasonable that these changes may be related. Even if they are not, it is arguable that automatic deduplication goes against the more weakly-worded remaining principle:

> Individuals can use one or more Identity Service Providers to maintain separate or merged personal and business Digital Identities. (*02 - Overview*, §2.3)

1.1 How EDIs are generated

The verified information about a user which is used to generate an EDI varies depending on the Identity Proofing Level (IP Level) required, as well as what information is available about the user. The information used is taken from verified documents (e.g., driver licences, passports, visas) which are deemed suitable for each IP Level; *Table 1: Documents used to build an EDI* [06, §2.3.1.2] lists the IP Levels and the associated documents. It is important to note that the EDI is generated according to the requested IP Level, not according to the IP Level that the user has achieved at the IdP.

If a verified document suitable for the requested IP Level is found, the EDI is generated by taking the document type code URN[27] and appropriate attributes for that document[28], concatenating them, encoding them in UTF-8, and hashing the resulting string using SHA-256 [06, FED-02-03-15 and FED-02-03-15b].

In the case that a user has not verified any of the acceptable documents, the IdP Link for the user and a globally-unique identifier for the IdP are concatenated, encoded in UTF-8, and hashed using SHA-256 [06, FED-02-03-15a].

1.2 How EDIs are used

The exact process for deduplication is not specified by the TDIF. The only guidance that it provides is a single, vaguely described possible implementation:

> One implementation of Deduplication is to transform the EDI into a unique identifier specific for a User at each Relying Party. This is then used as a lookup to check whether a different Digital Identity

[27] As specified in *Table 35: Document Type Code* [06D, §6.1]
[28] As specified in *Table 2: Document Attributes used to build an EDI* [06, §2.3.1.2]

with the same unique identifier has previously accessed that Relying Party.

If there is, the RP link of that Digital Identity is mapped to the IdP link of the User. This ensures that Deduplication isn't done across entire identities, but instead is done at each Relying Party and that an Individual can appear the same to a Relying Party, regardless of which IdP was used. This unique identifier can also be configured in accordance with Relying Party sector identifiers. (*06A - Federation Onboarding Guidance*, §2.2.1.2)

Appropriate transformations of the EDI are not discussed.

The TDIF does place several limitations on the use of EDIs:

- An EDI may not be sent to any entity other than the IdP from whom it originated and the IdX who requested it [06, FED-02-03-16 and FED-02-03-18]
- An IdX may not store an EDI [06, FED-02-03-17]
- An IdX may not use an EDI as a pairwise identifier (i.e., the RP Link) [06, FED-02-03-17]
- An EDI may only be used to deduplicate identities at a specific RP (or group of RPs with the same sector identifier) [06, FED-02-03-19]

Additionally, it requires that deduplication only occurs for identities which have been proved to the same IP Level [06, FED-02-03-11].

In order for deduplication to be successful, the EDI must clearly be stored in some way as in the provided possible implementation, although it is not allowed to be stored directly. The lack of supplied acceptable methods of transforming the EDI means it is entirely up to the discretion of the IdX, which leaves room for inappropriate choices such as simply hashing the EDI again. Indeed, it is not clear that an appropriately secure and practical implementation exists at all[29], given the conflicting requirements of fast matching against transformed EDIs that are also resilient to the attack described below.

2 Issues

The deduplication process and the EDIs it uses have a number of issues, which at best render the process ineffective in many circumstances and at worst potentially leak highly sensitive personal data.

2.1 Deduplication will fail (by design) in many circumstances

As specified in FED-02-03-11, deduplication may only be done across identities which have been proven to the same IP Level. The likely intent of this is to prevent an identity which has been proven to a low IP Level being equated to

[29] I do not, of course, claim that it is impossible.

one which has been proven to a high IP Level and thus placing higher trust in the lower IP Level identity than it should receive. The presence of this requirement is probably necessary in order to ensure the integrity of the deduplication process. However, it also significantly limits the circumstances in which deduplication can occur. For deduplication to consistently succeed, a user who wishes to use multiple different IdPs must ensure the level of proof at each IdP is the same at any given time.

Determining the IP Level to which an EDI corresponds is non-trivial, as the EDI generation process does not incorporate any data which ties it to a specific IP Level. Since a given document can be valid for multiple different IP Levels[30], this means that additional data is needed. The TDIF does provide a means for supplying this data in the form of Authentication Context Class Reference (ACR) values in the authentication request, which encodes the IP Level and Credential Level pairs that will satisfy the request. An IdX may include a list of accepted IP Levels for a given request, and the IdP must reject the request if the user's identity is not proven to the request IP Level.

While the TDIF does specify how ACR values should be requested and returned [06B, §3.8.3], it provides conflicting and unclear specifications. An IdX can request the ACR value as an essential claim or as a voluntary claim[31]. In the case that the ACR value is requested as a voluntary claim[32], OIDC-03-08-09 [06B] states that the IdP *MAY* return the ACR value that the user has achieved, i.e., that the IdP has full discretion as to whether or not to return it. This directly contradicts the earlier OIDC-03-07-15 and OIDC-03-07-24 [06B], which dictate that the IdP must always return the ACR value. While it's likely that the latter requirements are the correct ones, the TDIF provides no clarity on how conflicting requirements should be interpreted, which leaves the interpretation up to the discretion of the IdP and their assessor.

In the situation that an ACR value is not returned, there is no indication of the IP Level of the EDI. Because requesting an ACR at all is optional[33] and a request which includes any ACR values must include all ACR values that satisfy or exceed the requested value[34], the EDI's IP Level is ambiguous if no ACR value is included in the response. There is no specific indication of how this situation should be handled but the most appropriate interpretation is that deduplication using an ambiguous EDI should not be performed at all.

Even if the user has proved their identity to the same IP Level at different IdPs and the IP Levels of their EDIs are unambiguous, that's not necessarily

[30] For example, a birth certificate is valid for every IP Level greater than IP1.

[31] The details on how these are differentiated are provided in §2.8.3 instead, which is about RP–IdX requests and not relevant to the IdP. §3.8.3 does not reference §2.8.3; instead, it just makes reference to the concepts of essential and voluntary claims and assumes that the reader understands.

[32] Which is the case in all observed real-world requests.

[33] OIDC-03-06-06 specifies that the requested values should be mapped from the RP–IdX request, and OIDC-02-06-05 specifies that they are optional in that request [06B].

[34] The original RP–IdX request may include only a single ACR value, which is then transformed into the full list of acceptable values in the IdX–IdP request [06B, OIDC-03-08-06].

sufficient to guarantee that deduplication can be performed. The document used to generate an EDI is selected from the list of valid documents for the IP Level by taking the first verified document for that user according to the order set out in *Table 1: Documents used to build an EDI*. This means that if a user verifies different documents at different IdPs—a situation which does not seem unlikely, should a user set up their accounts at different IdPs at different times—then the EDI returned by each IdP for the same IP Level may still differ as they are derived from different documents.

For example, Alice achieves IP Level 2 by verifying two documents at each IdP: her passport and driver licence at IdP A and her driver licence and Medicare card at IdP B. She uses an RP with her IdP A identity and then her IdP B identity. Even though Alice has IP 2 at both IdPs, the IdX cannot successfully determine that her identities at IdP A and IdP B are the same person, as the EDI from IdP A is derived from Alice's passport while the EDI from IdP B is derived from her driver licence.

Even when a user has no verified documents, the TDIF still defines a process for deriving an EDI [06, FED-02-03-15a]. It's unclear when a valid interaction in the TDIF might involve a user with no verified documents at all, as even a verified email address or mobile phone number is sufficient to achieve IP Level 1. Even if the situation does arise, an EDI generated in this way is seemingly useless for deduplication. Because it is generated from the IdP Link and a unique identifier for the IdP, both of which are specific to the IdP in question, EDIs generated in this way for the same user at different IdPs will always be different[35].

As is typical for the TDIF, no justification is provided for why deduplication is defined in this circumstance at all. If the deduplication process is always intended to fail in this case, it would be more reasonable that the IdP returns no EDI at all and the IdX does not attempt deduplication if no EDI is returned. Not returning an EDI in the case that no meaningful EDI can be generated is consistent with existing requirements which allow an IdP to selectively exclude claims from its response [06B, OIDC-03-07-19]. While this does not increase the number of cases in which deduplication will succeed, it makes it more clear that deduplication is intended to fail and it is simpler.

2.2 Not all specified document types can be used

Requirement FED-02-03-15 states that EDIs must be generated by concatenating the document type code URN and some document attributes. However, not all documents actually have a valid URN associated with them in *Table 35: Document Type Code* [06D, §6.1], despite being indicated as valid documents.

- *Table 1: Documents used to build an EDI* [06, §2.3.1.2] indicates that a verified email address or mobile number is considered a "document" suitable for meeting IP Level 1, but neither of these have URNs. These two documents are the only valid documents for IP 1.

[35] Short of a hash collision, the probability of which is negligible.

- A passport is a valid document for IP 1 PLUS, IP 2, IP 2 PLUS, and IP 3, but also has no associated URN.

Since these documents have no associated URN, it's impossible to correctly generate an EDI from them. It's probable that not including these URNs is merely an oversight, but it nevertheless renders them unusable because there is no guarantee that different IdPs will generate them the same way.

2.3 An attacker can use EDIs to learn highly sensitive information

An EDI is derived from information based on a single document. A single document contains a very limited amount of entropy: for example, the specified attribute for a passport has approximately 30 million valid values. Because the EDI generation process incorporates no additional secret source of entropy, it is therefore trivially possible for an attacker to derive the information from which an EDI was generated, and thus to learn highly personal data such as a passport number or a Medicare number.

There are two key attack scenarios:

1. An attacker has knowledge of a single EDI (e.g., from a leaked ID token) and wishes to learn the information from which it was derived. The attacker may or may not have knowledge about the person with whom the EDI is associated, such as the person's date of birth.
2. An attacker compromises an IdX and gains access to the table of transformed EDIs used for deduplication in the manner suggested by the TDIF and wishes to learn the original information en masse.

Attacking a single EDI

A single EDI can be trivially brute forced by an attacker with sufficient computing resources. An EDI is derived entirely from public fixed data (e.g., document type code URNs) and one or more known sources of entropy (i.e., the document attributes). Notably, there is no source of entropy other than the document attributes. This means that the search space for the EDIs derived from a given document is only as large as the product of the sets of possible values for each attribute for that document. For example, a passport only uses a single attribute with approximately 30 million valid values ($\sim$25 bits of entropy) so there are only 30 million possible passport-based EDIs. The attributes for a Victorian birth certificate give $\sim$32 bits of entropy (a 5-digit registration number gives $\sim$17 bits, a date of birth in the last 80 years gives $\sim$15 bits of entropy, and the state gives no additional entropy as there is only a single valid value).

Because only a single document is used to derive the EDI, the total search space for all EDIs is calculated additively. For example, calculating all EDIs derived from passports and VIC birth certificates is approximately 1% more difficult than calculating just those derived from VIC birth certificates. This means that the large number of usable documents does not significantly increase the difficulty of calculating all possible EDIs, as an attacker need only calculate each

Documents	Number of values[37]	Time (s)
Passport	$\sim 3 \times 10^7$	4
Driver licence (VIC, WA, SA)	$\sim 1.02 \times 10^9$	185
Birth certificate (VIC)	$\sim 3 \times 10^9$	540

Table 5.1: Measured times to derive EDIs for all possible combinations of attributes.

document's EDI individually, rather than having to calculate all combinations of possible documents.

To determine the feasibility of calculating all possible EDIs, I wrote a proof-of-concept program in Rust. It generates all possible combinations of attributes for a subset of the documents from which EDIs may be derived and calculates hashes of those attributes as specified by FED-02-03-15 and FED-02-03-15b, searching for a match with a specified value and returning the original document. The program uses rayon[36] to parallelise the calculations, but is unoptimised. Times for some of the runs on a consumer desktop computer with a 4-core 4GHz Intel i7-6700k CPU can be seen in Table 5.1.

It is clear from these results that attacking EDIs in this way is feasible. While some documents have much greater entropy (such as ACT birth certificates, which have ~42 bits, or $\sim 3 \times 10^{12}$ potential values for an expected runtime of approximately 150 hours on the same consumer desktop computer), the availability of large amounts of compute power on-demand through cloud computing providers such as AWS means even these documents are still easily feasible for a determined attacker. It is trivial and cheap for an attacker to gain access to hundreds of cores of compute for only as long as needed to attack EDIs. Obtaining 100 CPUs on AWS EC2 and calculating all ACT birth certificate-based EDIs would take approximately 6 hours and cost US$2.94 at current pricing[38].

If the attacker knows the identity associated with the EDI[39], they may be able to make use of additional data to make brute forcing the EDI easier. For example, knowing the date of birth of the user reduces the search space of all birth certificates, citizenship certificates, and visa by ~15 bits—this would reduce the expected 6 hours on 100 EC2 CPUs to less than 20 seconds. Additional information such as the IP Level of the EDI further limits the search space to only a subset of the possible documents.

Even if any of the documents has sufficient entropy available in its attributes that it is not feasible to brute force all EDIs generated using it, an attacker can just ignore that document type and brute force all of the others, because an EDI

[36] https://github.com/rayon-rs/rayon

[37] Due to difficulty finding exact formats for many of these numbers, these values are estimates based on the best available knowledge.

[38] Using EC2 Spot Instances (https://aws.amazon.com/ec2/spot/pricing/). On-Demand Instances would cost US$15.30 for the same time (https://aws.amazon.com/ec2/pricing/on-demand/).

[39] For example, if the EDI is obtained through something like a malicious browser extension which exfiltrates TDIF ID tokens when it detects logins to a TDIF IdP.

is only generated from attributes belonging to a single document.

While these calculations are at best only rough estimates, they give a strong indication that it is very feasible for an attacker with only moderate resources to derive the information on which a single EDI is based through a brute force attack.

Attacking the table of transformed EDIs

Attacking a large number of transformed EDIs is a simple extension of attacking a single EDI. While the TDIF disallows an IdX from directly storing EDIs, it directly encourages storing a transformed version, although it does not specify allowable transformation functions, and thus the most difficult part of attacking a table of transformed EDIs is determining what transformation function was used. Hiding the transformation function is not an adequate security mechanism to protect the table of transformed EDIs:

> System security should not depend on the secrecy of the implementation or its components. (§2.4, *Guide to General Server Security*, NIST)

Any or all of leaks, insider knowledge, intuition and understanding of the domain, or poor security practices can make it trivial for an attacker to determine what transformation function has been used.

In the case that the attacker already knows the transformation function, attacking the table is as simple as extending the EDI generation from the single EDI attack to additionally transform each generated EDI. For each EDI generated, the attacker searches the table for matches and records them, ending when the search space is exhausted or all entries in the table have been matched. The choice of transformation function affects how much slower this is than the single EDI attack.

While it's technically possible to pre-compute the transformed EDIs (or pre-compute the EDIs and perform just the transformation for the full attack to avoid having to store a copy for each transformation), given how easy EDIs are to derive and the storage space requirements for multiple trillion entries, there is probably little reason to do so. One possible trade-off of space in favour of time would be to generate and store the EDIs for just the suspected "most likely" documents (probably passports, driver licences, and Medicare cards) to reduce the time needed to calculate just the most common EDIs, but this probably depends on what information the attacker deems most valuable.

Improving the naive attack

The proof-of-concept used here takes a naive approach to generating document attribute combinations and assumes that all potential combinations are equally valid. However, an intelligent attacker will realise that this is not necessarily the case. Many attributes are only valid for a subset of their domain, which means a significant portion of the search space can immediately be eliminated. For example, while a Medicare card number appears to be a 10-digit number ($\sim$33 bits of entropy), it actually consists of an 8-digit identifier whose

first digit should be in the range 2–6, a 1-digit checksum, and a 1-digit issue number[40]. Assuming all issue numbers are equally valid[41], these restrictions reduce the number of possible values to 5×10^7 ($\sim$26 bits of entropy), which is only 0.5% of the original search space. An attacker with good knowledge of the formats of the documents used to generate EDIs can take advantage of this to significantly reduce the search space for EDIs, further increasing the viability of attacking them.

An attacker may also decide to attack only a subset of the documents which are available, because certain document types will likely account for a majority of EDIs—fewer people will be verified to higher IP Levels than those who achieve lower IP Levels, so documents acceptable at lower levels (such as passports, driver licences, or Medicare cards) are likely to be more frequently encountered and so an attacker with less resources may achieve a majority of the benefit with a small amount of the effort.

3 An alternative approach

The documents used to generate EDIs were not designed with this use-case in mind, and so their attributes do not contain high enough entropy to be safely used in this manner. As such, it is clear that the TDIF should not use these documents in such a manner; this was reflected in the report to the DTA, which recommended immediate deprecation and eventual removal of the existing process.

In this section I propose instead an alternative approach which does not rely on any personal user data, instead suggesting a simple, low-friction manual step to the normal authentication process which allows a user to explicitly opt to link their digital identities. The exact mechanics of this approach are not fully explored; instead, it is intended as a basis from which a more fully considered approach might be derived.

Manual deduplication of user accounts

A manual deduplication process which respects user choice and informed consent may look something like the following:

1. A user logs into their intended IdP and is redirected back to the IdX to give consent for attribute sharing.
2. If the IdP Link returned has not been encountered by the IdX before, the IdX presents the user with a prompt: *Have you logged in to MyTaxSystem using a different identity provider before?*
   (a) If the user selects no, the transaction is complete.
   (b) If the user selects yes, they continue.

[40] https://www.clearwater.com.au/code/medicare

[41] Which is not the case, but knowledge of the cardholder's date of birth is required to reduce this further.

3. The IdX allows the user to select any IdPs they have used before to merge identities at the RP[42].
4. For each selected IdP, the IdX initiates a login process. If it successful and meets the other criteria for deduplication, the IdX joins the identities in its records.
5. The user is returned to the RP, which sees the RP Link of the previously used IdP.

While this process requires additional manual work for a user, it is consistent with the principle of allowing user choice, and it does not require any personal information other than that which would be returned in a standard login process. The IdX is required to store a mapping between IdP Links and RP Links [06, FED-02-03-04][06A, §2.2.1.1], so joining accounts at different IdPs simply involves mapping the different IdP Links to the same RP Link. Should an attacker gain access to the table storing this mapping, this change would allow them to learn that the different IdP Links belong to the same user, but they learn nothing more than that[43]. This is clearly significantly more acceptable than a potential attacker learning poorly-disguised, sensitive personal data about the user.

Potential solutions which should be avoided

The approach suggested above is clearly a significant deviation from the approach specified in the TDIF. As a result, the DTA may be inclined to avoid such a rapid change and instead seek alternative approaches which fit better with the existing specification. However, not all such alternatives are appropriate.

Using more than one document as an attribute source

Solutions such as combining attributes from multiple documents may seem promising, as increasing the number of source documents increases the size of the search space multiplicatively with the number of documents, as opposed to the additive scaling of the current process. However, this is still susceptible to attacks where the attacker knows some information about the owner of the EDI and can use it to reduce the search space to a feasible size. This is of particular concern because verified documents are an attribute which can be requested by an RP in the TDIF [06D, §3.1.4], which increases the possibility that an attacker may have access to relevant knowledge about the owner of an EDI.

[42] With an appropriate description of what this entails, and a note that if they need only select one IdP that they have used before (since all IdPs that they've deduplicated in the past will share the RP Link, using the same RP Link as any one previous IdP joins the new IdP Link to all IdP Links stored against that RP Link).

[43] Assuming IdP Links are correctly generated, learning the IdP Link should give an attacker no knowledge about the user to whom it belongs or the IdP who generated it.

Higher entropy attributes

The use of attributes from identity documents which contain more inherent entropy may also appear to have potential. The first issue with this approach is finding such attributes: identity documents are typically designed for human use first and computer use second, and for the purpose the TDIF uses their attributes for, not at all.

However, even if we find an attribute which appears to have sufficient entropy, such as a passport photo, it doesn't solve all the problems. It's immediately clear that the issue of EDIs derived from different documents failing to match remains, as even attributes containing high-entropy data will differ across different documents. Further, most of the attributes used to derive EDIs remain static across the document owner's lifetime, which makes the attribute agnostic of document expiries and updates. With attributes such as passport photos, however, this is not the case, which adds the issue of handling EDI staleness.

Additionally, very few identity documents have high-entropy attributes at all, which limits the documents that can be used to derive EDIs. It seems unlikely that the set of documents that can be used to establish Identity Proofing levels will be reduced, so this reduces the likelihood that an EDI can be derived for a user at all, which again makes automatic deduplication of little use.

Alternative sources of entropy

It is clear that deriving EDIs from attributes sourced from identity documents is not an intended use-case of the documents, and nor is it secure. The next likely solution might perhaps appear to be finding an alternative source of entropy from which to derive EDIs. However, it is not immediately clear that an entropy source exists which is sufficiently constant (i.e., it does not change over time), universal (i.e., enough people have it that deduplication will in most cases succeed), and private that it would be appropriate to use for this purpose. As a result, it seems unlikely that attempting to find such a source of entropy is a better solution than using a non-automatic approach.

Chapter 6

The TDIF protocol

The TDIF specifies the protocol which must be used by the RP, IdX, and IdP to carry out the authentication process. It describes two variants of the protocol, one of which uses OpenID Connect (OIDC) 1.0[44] [06B, 39] and the other of which uses the Security Assertion Markup Language (SAML) v2.0 [6, 06C]. In TDIF v1.5, it was explicitly indicated that OIDC is the preferred standard and SAML is included primarily to allow easier integration with existing SAML-based services.

> The preferred federation standard for the TDIF identity federation is OpenID Connect 1.0 (OIDC). OIDC is based on a modern collection of standards that simplify the technical integration for Relying Parties and Identity Service Providers.
>
> [...]
>
> At the time of writing, SAML 2.0 is still widely used in federated identity solutions, so a SAML 2.0 profile will be provided that is functionally equivalent to the preferred OIDC-based technical integration standards. (*TDIF Architecture Overview*, §2.5.5)

OIDC is also used by myGovID, and so this work limits its scope to the OIDC profile. However, it is notable that TDIF Release 4 no longer includes this statement, although it does only require that IdXs implement OIDC and leaves SAML as an optional requirement [06, FED-02-01-01, FED-02-01-02], which suggests that OIDC is still the preferred protocol. Analysis of the SAML profile is left as future work.

In this chapter, I provide an overview of the OpenID Connect protocol on which the TDIF is based, as well as OpenID Connect in published literature. I then outline how the TDIF builds on OIDC to create its two-legged protocol specification. Finally, I observe the observed protocols of myGovID and Digital iD and discuss a number of deviations from the specified protocol found therein.

[44] https://openid.net/connect/

1 OpenID Connect 1.0

The OpenID Connect (OIDC) protocol builds on OAuth 2.0[45] [8], which itself is a widespread protocol used to provide authorisation to HTTP services on behalf of resource owners. OIDC expands OAuth 2.0 to provide a formal means by which services can request identity verification, authentication, and information about users. It uses JSON Web Tokens (JWTs) [37] to convey information between participants, and provides signing and encryption of JWTs using JSON Web Signature [36] and JSON Web Encryption [23] respectively.

1.1 The protocol in brief

In the following description (and the rest of this chapter), the TDIF term Identity Provider (IdP) should be considered equivalent to the OIDC term OpenID Provider (OP). The TDIF term Relying Party (RP) should be considered equivalent to the OIDC term Relying Party; because an RP also always fulfils the OIDC role of Client, these terms should be considered interchangeable[46].

OIDC defines three modes of operation: the authorization code flow, the implicit flow, and the hybrid flow. The TDIF specification dictates the use of the authorization code flow, so that is the mode outlined by the following description. The differences between the authorization code flow and the implicit flow are described afterwards, while the hybrid flow is considered out of scope of this work.

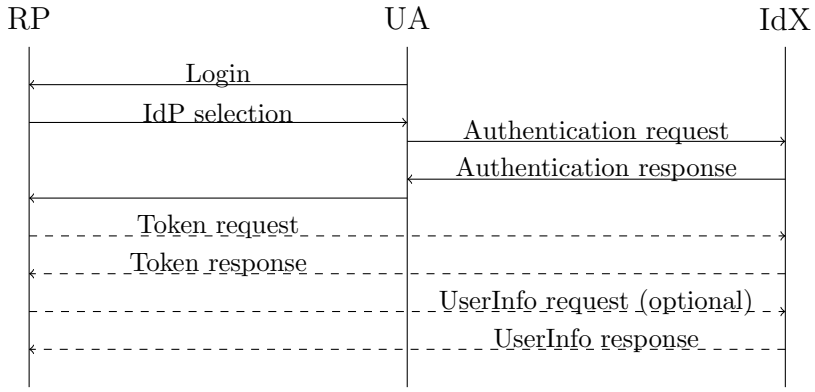

Figure 6.1: The OpenID Connect 1.0 authorization code flow. Dashed lines represent server-to-server requests.

The OIDC authorization code flow, shown in figure 6.1, executes the following steps:

[45] https://oauth.net/2/

[46] There is a slight distinction in that an RP can be one of multiple different types of client, but for our purposes and for the sake of consistent terminology, this distinction is ignored.

1. **RP prepares an authentication request**
   The authentication request is an OAuth 2.0 authorization request. It uses either HTTP `GET` (sending parameters in the URI query string) or `POST` (sending parameters as the request body using Form serialization). A valid OIDC authentication request requires the following parameters:

   scope
   MUST contain the value `openid`, and MAY contain other scope values. The TDIF defines a number of additional scopes [06D, §4.1].

   response_type
   This value indicates which mode is used: `code` indicates the authorization code flow, while `id_token` or `id_token token` indicate the implicit flow.

   client_id
   An OAuth 2.0 identifier for the RP sending the request.

   redirect_uri
   The URI at the RP to which the user agent should be sent once authentication is complete. In the TDIF, this must be an absolute URI protected by TLS (i.e., an HTTPS URI).

   state
   A value which is used to mitigate Cross-Site Request Forgery (CSRF) attacks. This parameter is only recommended in OIDC, but is required in the TDIF.

   nonce
   A value containing sufficient entropy to prevent guessing, returned to the RP in the token responses and used to mitigate replay attacks. This parameter is optional in OIDC in the authorization code flow, but required in the implicit flow and in the TDIF.

2. **RP sends the authentication request to the IdP's Authorization Endpoint**
   The RP sends its prepared request to the IdP over a TLS-protected connection, where the IdP validates the request.

3. **IdP authenticates the user**
   The IdP authenticates the user through some means which are unspecified by OIDC.

4. **IdP obtains user authorisation**
   Once the user is authenticated, the IdP is required to obtain authorisation from the user to release information to the RP.

5. **IdP sends the user back to the RP with an authorization code**
   Once authorisation is obtained, the IdP returns the user agent to the RP at the endpoint indicated by the `redirect_uri` parameter in the authentication request. This response includes an OAuth 2.0 authorization code [8,

§4.1.2], which in the TDIF is defined to be a random string [06B, OIDC-02-07-12], and the `state` value from the original request.

6. **RP sends a token request to the IdP's Token Endpoint**
   The RP then redeems this authorization code at the IdP's token endpoint for an ID token and an access token. This request happens between the RP server and the IdP server—it is not visible to the user agent. This request uses HTTP `POST` and includes the following parameters:

   grant_type
   Must be set to `authorization_code`.

   code
   The authorization code received from the IdP.

   redirect_uri
   The same URI used for the original authentication request.

7. **RP receives a response containing an ID token and access token**
   The RP receives a JSON response containing two tokens. The ID token, stored under the key `id_token`, is a JWT, while the access token, stored under the key `access_token`, is of an unspecified format. The response also includes a token type (which must be `Bearer`), the token lifetime in seconds, an optional refresh token[47], and the scope of the access token.

8. **RP validates the ID token and retrieves the user's identifier**
   The RP must finally validate the ID token received from the token endpoint, which involves decrypting the token if encrypted, validating its signature if it is signed, checking the issue and expiry times, verifying the nonce against the original value if it was present in the authorization request, checking the issuer (who created the token) and audience (who the token is destined for—the RP), and a number of other checks. If the ID token passes validation, the RP can finally extract the `sub` value—the subject identifier, or the identifier by which the IdP identifies the user.

Assuming this process is complete, the user is now authenticated at the RP. The RP can optionally use the access token it received to request *claims* from the IdP's UserInfo endpoint, which are attributes containing information describing the user (e.g., their name or email address).

Differences in the implicit flow

The implicit flow follows the same steps as the authorization code flow (except for its `response_type` value in the authorization request) until the point at which the IdP would redirect the user back to the RP with an authorization code (step 5 above). Instead, the IdP sends the ID token and possible access token back to the RP directly via the user agent, sending them in the fragment component of

[47] Refresh tokens can be used to obtain new access tokens; their use is outside the scope of this work.

the redirect URI[48]. Validation of these tokens is then much the same as in the authorization code flow.

1.2 Signing and encryption

OIDC specifies the use of both digital signatures and encryption, although both are optional to use. Both symmetric and asymmetric primitives are supported; however, as the TDIF uses asymmetric primitives, we will consider only those.

Any value which is sent as a JWT may be signed using JWS [36] and/or encrypted using JWE [23]. If it is both signed *and* encrypted, it must be signed first, then the resulting JWT is encrypted. When a JWT is signed or encrypted, it will contain a header known as a JOSE header [37, §5] which indicates the algorithm used and some identifier for the cryptographic key.

Keys distribution is not specified by the TDIF. However, an IdP might distribute its public keys via a URI in the form of a JSON Web Key Set (JWKS) [22], the location of which is advertised in its Discovery document [40], a publicly-available JSON document at a constant endpoint (the OIDC server base URI followed by `.well-known/openid-configuration`). Advertising its public key in this way both simplifies the onboarding process and allows for easy key rotation.

1.3 Request objects

Instead of passing authentication request parameters as query string parameters, it is possible for the request to instead pass them as a JWT referred to as a *request object*. Passing the request in this manner allows the parameters to be signed and/or encrypted, which allows for client authentication and confidentiality of the request parameters. Request objects can be sent in one of two ways: directly in the request via the `request` parameter; or indirectly via the `request_uri` parameter, which contains a URI from which the IdP can then retrieve the request object.

Even when request objects are used, the `client_id` and `response_type` parameters must be sent normally to ensure that the request is valid in OAuth 2.0.

2 TDIF protocols in literature

As prominent standards used increasingly widely in industry, both OpenID Connect and the underlying OAuth 2.0 have seen significant analysis in literature. While a number of attacks identified in these works have since been fixed or mitigated, some still apply to the TDIF.

[48] Other response modes can be requested in the original authorization request instead, but using the fragment is the default.

In [16] Fett et al. formally model OAuth, including the base OAuth 2.0 standard [8] as well as additional RFCs and web best practices to ensure their model represents securely-implemented OAuth. Using this model, they identify three key security properties of OAuth: authorisation; authentication; and session integrity, which is divided into session integrity for authorisation and session integrity for authentication.

Authorisation
: An attacker is not able to access or use protected resources which are available to an honest RP at an IdP without the user's browser or IdP being corrupted in some way.

Authentication
: An attacker cannot log in at an honest RP under the identity of a user, again without the user's browser or IdP being corrupted.

Session integrity for authorisation
: a) an OAuth flow is completed with a user's browser (that is, the RP gets access to a protected resource of the user's identity) iff the user started an OAuth flow, and b) if the IdP in the completed flow is honest, then the identity for which the flow was completed is the same as the identity for which the user started the OAuth flow.

Session integrity for authentication
: a) a user is logged in with some identity iff the user started an OAuth flow, and b) if the IdP in the flow is honest, then the identity under which the user is logged in at the completion of the flow is the same identity for which the user started the OAuth flow.

The authors, using the same model, identify four attacks on correctly implemented OAuth 2.0. In the 307 Redirect Attack, a malicious RP takes advantage of the lack of specified HTTP redirect status code for redirecting the user agent back to the RP to learn a user's credentials at an RP which redirects using HTTP 307 Temporary Redirect, which resubmits the form if the preceding request was a form submission POST request. In the IdP Mix-Up Attack, a network attacker modifies requests between the user's browser and an RP to convince the RP that the user wants to log in using an attacker-controlled IdP instead of the honest IdP the user wants to use. This allows the attacker to trick the RP into sending its IdP an authorization code issued by the honest IdP, which the attacker can then redeem at the honest IdP. In the State Leak Attack, an attacker whose resources are used at the RP's redirection endpoint can learn an authorization code belonging to the victim through the HTTP Referer header, which the attacker can then use to either authorize themselves at the RP as the victim or to log the victim in as the attacker at the RP. In the Naïve RP Session Integrity Attack, an attacker-controlled IdP tricks a naive RP into thinking that a user logged in at an honest IdP, instead logging in at that RP as the attacker.

The authors also suggest mitigations for each of these attacks. Some of these, such as the use of HTTPS for a number of key steps in the authorization

process to mitigate the IdP Mix-Up Attack, are almost ubiquitous in modern applications, while others are not widely known or hidden in since-expired IETF drafts [38].

Fett et al. [15] expand on Fett et al. [16], applying the same approach to OpenID Connect 1.0. Using their formal FKS model of OIDC, they identify the same four security properties as they did for OAuth: authorisation, authentication, session integrity for authorisation, and session integrity for authentication. Additionally, they identify a number of secondary security properties specific to OIDC. The authors summarise and provide mitigations for known attacks on OIDC and, by taking these into account in their FKS model of OIDC, prove that OIDC is secure with regard to the primary and secondary security properties they identified.

Some of the mitigations that they suggest (such as use of referrer policies to prevent leaks of state) are not formalised as part of OIDC itself, which means that an implementer (RP, IdP, or OIDC library) may not implement these mitigations. Others were published as IETF drafts which have since expired with no obvious replacement [21, 24, 38]. Yet others are published as mitigations for attacks against OAuth. Because not all attacks which affect OAuth also affect OIDC, it's not necessarily immediately clear whether such a mitigation is necessary for an implementer of OIDC. Their model nevertheless operates under the assumption that these mitigations are in place; while it may formally prove the security of the OIDC spec implemented correctly and with appropriate awareness of the many attack vectors and appropriate mitigations, it is unlikely many real-world implementations of OIDC meet this requirement.

Li et al. [25] consider Cross-Site Request Forgery (CSRF) attacks against OAuth 2.0 (and by extension, OpenID Connect). They outline existing CSRF defences, concluding that the only defence which applies to OAuth is the use of a secret validation token—in OAuth, the *state* parameter [8, §10.12]—as header-based defences (such as the HTTP Referer and Origin headers) do not work effectively against cross-origin requests, which is the usual case for IdP-to-RP post-login redirects. The authors claim that many implementers of OAuth misuse the *state* parameter, so they propose a novel CSRF mitigation technique using which combines the HTTP Referer header with user intention tracking based on registering different redirect URIs for different IdPs (referred to by Fett et al. [16] as naive user intention tracking). In their proposed solution, an RP can determine if a redirect is legitimate by extracting the expected IdP from the redirect URI to which the request was made and checking it against the value of the Referer header. Because this header is generated by the victim's browser, it is very difficult for an attacker to spoof to a legitimate value. The authors note that this defence also applies to RPs which use explicit user intention tracking, in which case the only difference is that the RP extracts the intended IdP from the user's session state rather than the redirect URI.

While the authors discuss some limitations involving browsers automatically

suppressing the Referer header in some situations, one aspect they do not consider is users who value privacy (an increasingly large proportion of users) explicitly disabling Referer headers in their browser, either through browser settings or third-party addons. In the authors' solution, this would likely result in these users being entirely unable to access RPs which use this mitigation technique. If they were to weaken this to allow requests which have no Referer header at all, an attacker can then suppress the Referer header in the request, thus avoiding the mitigation. A potential improvement to their mitigation is to preferentially check the Origin header instead of the Referer header, as it is less likely to be suppressed due to privacy concerns. This can also be combined with the Access-Control-Allow-Origin header to allow only requests whose Origin matches the IdP expected for the redirect URI.

It seems likely that this defence would also adequately mitigate the naive RP session integrity attack outlined in [16], which is effectively a CSRF attack from another IdP at the same RP.

Mainka et al. [26] analyse OpenID Connect using a malicious IdP to manipulate the messages sent from the IdP to the RP. Using this approach, they categorise both known and novel attacks into one of two categories: *Single-Phase Attacks*, which are attacks which can be carried out by manipulating a single point of the SSO process; and *Cross-Phase Attacks*, which are more complex attacks which target multiple messages at different points in the protocol. The authors identify two novel Cross-Phase Attacks, both of which rely on the RP supporting the optional OpenID Connect Discovery extension [40]: an IdP Confusion attack which closely resembles the IdP Mix-Up attack against OAuth 2.0 described by Fett et al. [16] and allows a malicious IdP to log into a vulnerable RP as the victim user; and a Malicious Endpoints attack, in which a malicious IdP provides a discovery document which mixes URIs from a legitimate IdP in with its own such that a vulnerable RP will send some messages in an authentication process to the honest IdP and some to the malicious IdP, leaking an authorization code and client secret for the honest IdP to the malicious one and thus allowing it to both log in as the victim user and impersonate the RP at the honest IdP. The authors also create an analysis tool called PrOfESSOS which provides automated evaluation of OIDC RPs to detect the attacks described in this work. The tool allows an administrator of an RP to configure and run tests against their RP to determine if it correctly implements mitigations against these attacks; it is the first tool which provides such a service for OpenID Connect.

The attacks described in this work apply to the TDIF as they do to any OIDC implementation, and if an RP in the TDIF ecosystem is vulnerable this may allow an attacker to register a malicious IdP at the RP and thus log in as a user of a TDIF system. While the authors do provide the PrOfESSOS tool as a service usable by the public[49], it requires that the RP opts into its use to prevent illegitimate users to identify potentially vulnerable RPs, which prevented it from being used as part of this work to assess RPs in the TDIF.

[49] https://openid.sso-security.de/rp-verifier.html

3 OpenID Connect in the TDIF

The TDIF expands on the base OpenID Connect specification in two main ways. First, it adapts OIDC to work with the TDIF's two-legged brokered model by using two nested OIDC authentication interactions, one between the RP and IdX and the other between the IdX and IdP. Secondly, it applies more strict requirements to the use of OIDC to improve security and limit what participants need to implement. It bases its stricter requirements on the International Government Assurance (iGov) Profile for OpenID Connect 1.0 [18], which itself is based on the iGov Profile for OAuth 2.0 [35].

3.1 Two-legged OpenID Connect

OpenID Connect is designed for a direct relationship between the RP and IdP, and so cannot be used as-is in the TDIF due to the presence of an IdX between the RP and IdX. The TDIF instead defines a two-legged alternate version of OIDC which breaks an authentication process into two halves, as shown in figure 6.2.

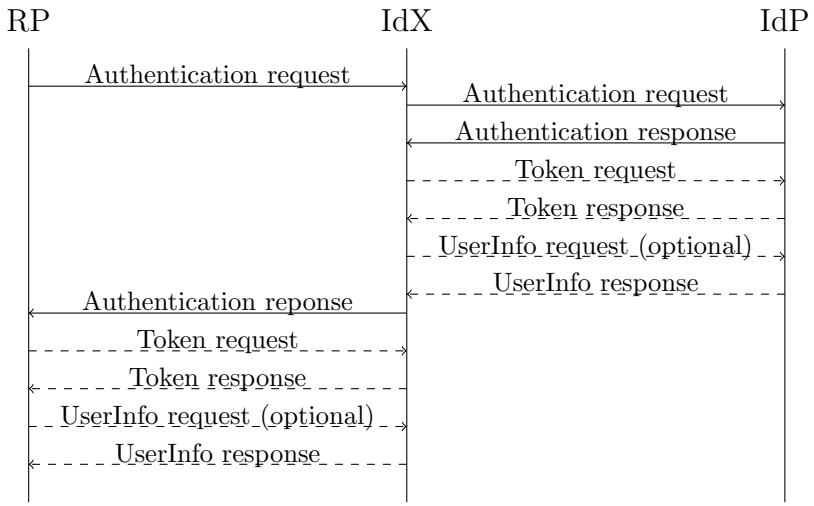

Figure 6.2: The TDIF two-legged OIDC variant. Dashed lines represent server-to-server requests.

In the first half, the RP acts as an OIDC RP and the IdX acts as an OIDC IdP. The second half begins immediately after the IdX receives the authentication request from the RP; at that point the IdX, now acting as an OIDC RP, modifies the authentication request it received to appear as if it were the originator of the request and sends that authentication request on to the IdP, which acts as an OIDC IdP. The second half proceeds as a normal OIDC authentication process, with the user authenticating themselves to the IdP, which returns an authorization code to the IdX which then redeems it for an ID token. From

that point, the first half resumes with the IdX gathering consent from the user to release attributes to the RP. The first half then itself proceeds as a normal OIDC authentication process, with the exception that the IdX uses the ID token returned from the IdP to authenticate the user.

3.2 Restrictions based on iGov

The iGov OIDC profile restricts implementers to a subset of the OIDC specification, and the TDIF profile follows suit. In particular, the iGov and TDIF profiles require the use of the authorization code flow [06B, OIDC-02-01-02] and a nonce [06B, OIDC-02-06-05], enforce client authentication in the request to the token endpoint [06B, §2.6.2, §3.6.3], and mandate signing of JWTs [06B, OIDC-02-07-15].

4 Observed protocols

For each of the existing accredited IdPs in the TDIF, I observed the implemented protocol to compare it against the TDIF specification. This was done by initiating an authentication process at an RP integrated with each of the IdPs and recording the observed network traffic using a proxy. Because OIDC is an HTTP-based protocol, this was simple to do. However, because the authorization code flow in OIDC involves direct server-to-server communication to redeem the authorization code for an ID token, this process was not observed and is thus outside the scope of analysis.

Because the TDIF protocol involves many steps and a lot of data, this section focusses on the deviations from the TDIF specification that were observed, rather than a comprehensive description of the observed protocols. Where not otherwise noted, the observed protocol matched the TDIF specification.

4.1 myGovID

myGovID was used for logging into the Relationship Authorisation Manager[50] (RAM). The observed protocol appeared to reasonably closely follow the TDIF specification. Due to certain parts of the protocol being direct server-to-server communication, they are impossible to verify. However, a limited amount of information about the background communication can be inferred based on what is visible to the client, such as when codes have been redeemed.

The observed authentication process using myGovID involved three separate entities: `authorisationmanager.gov.au`, the RP; `mygovid.gov.au`, the IdP; and `auth.ato.gov.au`, which was inferred to act as the IdX. It's notable that the DTA indicates that the only accredited IdX is run by the Department of

[50] https://authorisationmanager.gov.au/

Human Services[51] and yet the IdX used by myGovID is instead hosted on an ATO website.

The messages followed the expected two-legged path:

1. `authorisationmanager.gov.au` requests an OIDC token from the Authorization Endpoint of `auth.ato.gov.au`
2. `auth.ato.gov.au` requests an OIDC token from the Authorization Endpoint of `mygovid.gov.au`. The details included in the requested token are the same as in the previous request, but the request uses a different `client_id`
3. `mygovid.gov.au` requires the user log in and enter a code from their myGovID app
4. `mygovid.gov.au` redirects (using an HTTP 302 Found) back to `auth.ato.gov.au` with an Authorization Code
   - `auth.ato.gov.au` redeems the code via a back channel request to the Token endpoint of `mygovid.gov.au`
5. `auth.ato.gov.au` redirects (using a form POST) back to `authorisationmanager.gov.au`
6. The user resumes their task

While the observed protocol did adhere fairly closely to the TDIF specification, a number of deviations were observed, as described below.

RP-to-IdX authentication request: incorrect use of the implicit flow

The authentication request to the IdX's authorization endpoint used the parameters shown in Table 6.1. It is immediately observable that myGovID deviates from the TDIF specification: in the TDIF, this request is required to use the authorization code flow (`response_type=code`), but this request uses the implicit flow instead (`response_type=id_token token`).

RP-to-IdX authentication request: predictable state value

The TDIF recommends that the value of `state` should be an "unpredictable value with at least 128 bits of entropy" ([06B, OIDC-02-06-01]). However, this request uses a value of `OidcProviderType=AtoSsoIdp`, which is clearly not random[52]. Interestingly, in TDIF v1.5, this requirement was mandatory, not optional like it is in TDIF Release 4. It is not clear why it was weakened in this way.

RP-to-IdX authentication request: unknown `provider_hint` parameter

The request parameter `provider_hint` appears to be a custom parameter which suggests that a specific IdP should be used to service the request. It is defined

[51] https://www.dta.gov.au/our-projects/digital-identity/trusted-digital-identity-framework

[52] Or, perhaps, it is incredible luck. Who can tell?

Parameter	Value
redirect_uri	https://authorisationmanager.gov.au/myGovIdIsfOidcReturn
response_mode	form_post
response_type	id_token token
scope	openid profile https://authorisationmanager.gov.au/relationships email
state	OidcProviderType=AtoSsoIdp
nonce	48c7e405bcdb4a5fa2bbfe03ae39961b
client_id	https://authorisationmanager.gov.au
acr_values	urn:id.gov.au:tdif:acr:ip2:cl2 urn:id.gov.au:tdif:acr:ip1:cl2
provider_hint	https://ato.gov.au/myGovIdProvider

Table 6.1: Query parameters provided during the $RP \rightarrow IdX$ request

by neither OIDC nor the TDIF. Potentially, this is to request the use of an IdP that can handle the custom scope for the Authorisation Manager, https://authorisationmanager.gov.au/relationships. However, in general the IdX should make this decision by itself based on its knowledge of an IdP's understood scopes. Allowing the RP to request a specific IdP would seem to violate the integrity of the TDIF's double-blind property, as the RP may gain knowledge of which IdP is being used. It's unclear what would happen if the requested IdP were unable to service the request—if the IdX were to return a failure response when that IdP could not be used, then the RP can infer from successful responses which IdP is being used, but if the IdX chooses to continue with a different IdP, the RP may not gain any knowledge.

IdX-to-IdP authentication request: redirect doesn't reset HTTP Referer header

The response is a 302 Found redirect to a different page on auth.ato.gov.au, which redirects via several other pages on the same server before being redirected to mygovid.gov.au.

Because the original transfer from the RP to the IdX has no referrer policy set and the IdX transfers the user agent to the IdP using only redirects, the Referer header from the original request is preserved. This means the IdP receives a request containing an Referer header which clearly identifies the RP from which the request originates. This presumably violates the TDIF's double-blind property, as the IdP should not know the identity of the RP but can easily determine it from the HTTP Referer header.

IdP-to-IdX authentication response: unknown `session_state` parameter

The return from the IdP begins with a request to the endpoint at the IdX specified by `redirect_uri` in the original authentication request containing: the state parameter sent by the IdX originally; an OAuth 2.0 authorization code; and an undocumented parameter `session_state`, as seen in Table 6.2.

Parameter	Value
`code`	`1b6d25a55f51bb4b3d4f4e8e259ffd0b`
`state`	`3e89df814d87dfd53a3e59d6e6b4bc66`
`session_state`	`ecaTYmKm3eGRyujkQHwA3LIF9dPC0zvjo1-8p3Qw3p8.` `9d529ccb7d1866dca6f4e909acdee4ff`

Table 6.2: Query parameters provided during the $IdP \rightarrow IdX$ OIDC authentication response

The `code` and `state` parameters align with the TDIF and OIDC specifications. However, the `session_state` value appears to be a parameter specific to this implementation, as it is not defined in any of the specifications. It appears to consist of a base64-encoded value and a hex value concatenated with a period; however, the meaning of either of these values is unclear. Interestingly, this parameter was returned all the way to the RP in the authentication response from the IdX—this means it was passed (indirectly) from the IdP to the RP, and suggests that the RP has some knowledge of what it means.

A request for information about the purpose of this parameter was rejected by the ATO :

> Hi Ben,
>
> Thanks for your interest.
>
> To ensure the security and integrity of our online services, we are unable to provide any information about our source code.
>
> Regards,
>
> [name redacted for privacy]

The request asked only for any information about the purpose of the `session_state` parameter—it did not ask for source code.

Identical RP Link and IdX Link

The ID tokens received by the IdX and RP are shown in Listing 6.1 and Listing 6.2 respectively. In each of these JWTs, the `sub` field indicates the user pseudonym, which are the IdP Link and RP Link respectively. These values are very clearly the same—a clear violation of the TDIF requirement that the RP Link is not derived in any way from the IdP Link. If this behaviour is consistent

```
{
  "sub": "548524",
  "TokenType": "UserSignIn",
  "urn:ato:idType": "20",
  "session_id": "3e89df814d87dfd53a3e59d6e6b4bc66",
  "acr": "urn:id.gov.au:tdif:acr:ip1:cl2",
  "given_name": "Benjamin",
  "family_name": "Frengley",
  "birthdate": "1900-01-01",
  "email": "example@example.com",
  "email_verified": "true",
  "jti": "e17fe2ccf93c44989f59726c8f0ca62a",
  "iss": "https://ato.gov.au/bas",
  "aud": "https://ato.gov.au/bas",
  "exp": "1586075254"
}
```


Listing 6.1: OIDC JWT returned by the IdP to the IdX

across multiple RPs, it also allows colluding RPs to track user behaviour, as they will receive the same RP Link as each other.

Summary

Despite following fairly closely to the TDIF, myGovID deviates from a number of unambiguous requirements of the TDIF OIDC profile. Despite the use of the authorization code flow being specified in at least three different places in the TDIF profile [06B, OIDC-02-01-02, OIDC-02-01-06, OIDC-03-01-02] and in both the OIDC and OAuth iGov profiles, the RAM deviates from this by using the implicit flow. While the RAM acts as an RP in the recorded interaction, it is an accredited ASP and is communicating with an accredited IdX, both of whom have gone through the TDIF accreditation process, which is specifically intended to verify that accredited entities implement the appropriate requirements for their roles. Even if the RAM's interactions as an RP were not assessed at all during the accreditation process, the IdX should have been, so it is unclear why these inconsistencies with the TDIF profile are allowed by the IdX.

It's also interesting to note that the IdX involved in this interaction was potentially not the only acknowledge accredited IdX, as the DTA indicate that the accredited IdX is run by the Department of Human Services, while the IdX used by myGovID seems to be run by (or at the minimum, hosted by) the Australian Tax Office. It is unclear if this is merely due to a mistake on the DTA's website or a change of ownership of the IdX, or whether this IdX is an entirely different, seemingly unaccredited one.

```
{
  "iss": "https://ato.gov.au/myGovIdProvider",
  "aud": "https://authorisationmanager.gov.au",
  "exp": 1586057855,
  "nbf": 1586057255,
  "jti": "f267d3ef5ae742399ee1e13d088b2ca0",
  "nonce": "48c7e405bcdb4a5fa2bbfe03ae39961b",
  "iat": 1586057255,
  "at_hash": "dmDA0BMpZ4oqsQIqlDACjg",
  "sub": "548524",
  "auth_time": 1586057255,
  "idp": "https://ato.gov.au/myGovIdProvider",
  "acr": "urn:id.gov.au:tdif:acr:ip1:cl2",
  "given_name": "Benjamin",
  "family_name": "Frengley",
  "email": "example@example.com",
  "email_verified": "true",
  "birthdate": "1900-01-01"
}
```


Listing 6.2: OIDC JWT returned by the IdX to the RP

4.2 Digital iD

The Digital iD implementation of the TDIF was tested using the Australia Post Mail Hold service[53] as the RP. Partway through this process, Australia Post requests identity verification and offers the ability to do this using Digital iD. Authentication was aborted partway through the process, but late enough to observe that Digital iD does not seem to use the TDIF protocol.

From the very first request, it is clear that the Digital iD implementation strays from the TDIF specification. The initial request to Digital iD shown in Listing 6.3. This initial request would be expected to be an OIDC authorization request. However, the request shown above is clearly not a valid TDIF request, or even a valid OIDC request:

- `redirect_uri` is missing, which immediately invalidates the request in both the TDIF and OIDC protocols. An `origin_url` is present, but this is not a valid property in either OIDC or OAuth 2.0.
- `state` is missing. The use of this parameter is only *recommended* by OAuth2 and OIDC, but is required in the TDIF.
- `nonce` is missing, which is required in the TDIF.
- `scope` is missing, which is required in both TDIF and OIDC.

It is even more notable that this request goes directly from `auspost.com.au` to `digitalid.com`—it does not go via an IdX. All observed communication in this authentication process was directly between Digital iD and Australia Post, with no re-wrapping of requests to hide the identity of either participant. This is consistent with the ability of the Digital iD app to indicate to the user which

[53] https://auspost.com.au/mrso/mail-hold/

```
Host: digitalid.com
GET /oauth2/authorize?
  response_mode=query
  &response_type=code
  &client_id=ctid5VnE7Eb1jHinmVWIlOzA1Z
  &origin_url=https://auspost.com.au

302 Found
Location: https://digitalid.com/products/idv/login?
  continue=%2Fauthorize%3F
    partner_id%3Dptnr3u0iJFx4uNle7vA9iX5uzH
    %26response_mode%3Dquery
    %26response_type%3Dcode
    %26client_id%3Dctid5VnE7Eb1jHinmVWIlOzA1Z
    %26origin_url%3Dhttps%3A%2F%2Fauspost.com.au
```

Listing 6.3: The initial authorization request as observed in Digital iD.

RP sent the original authentication request[54].

From just these two pieces of information, it is clear that Digital iD does not implement the TDIF. Despite this, it is an accredited IdP under the TDIF[55]. The TDIF's accreditation process is designed to enforce that an accredited participant in the TDIF implements their role requirements under the TDIF correctly, and yet somehow Digital iD managed to become accredited without implementing either OIDC or SAML, the only two protocol profiles specified in the TDIF, and without using an IdX—the fundamental building block of the entire TDIF architecture and design. While the minor deviations from the specification observed in myGovID could be attributed to carelessness or incompetence on behalf of the Accreditation Authority, the total dismissal of the entire TDIF specification by Digital iD calls into question the integrity of the entire authentication process and the Accreditation Authority themselves. If the specification can be entirely ignored without being caught by the accreditation process, what is the value of either?

This complete failure of the accreditation process further exacerbates the privacy concerns related to the IdX discussed in chapter 4. The IdX has knowledge of almost everything that occurs in its identity sector and the only things preventing it from abusing this knowledge are the regulations set out in the TDIF—the same regulations which the accreditation process is designed specifically to enforce, and yet which are seemingly able to be ignored entirely. In the absence of the accreditation process—and more importantly, in the absence of integrity in the designers of the TDIF and enforcers of its regulations—there is no compelling reason why users should trust the TDIF at all.

[54] See subsection 4.1

[55] https://www.dta.gov.au/news/australia-posts-digital-id-accredited-under-tdif

Chapter 7

Code proxying attack on myGovID

In this chapter, I describe an attack on the login process used by the Australian Tax Office's myGovID IdP, which allows an attacker posing as a legitimate RP to secretly and indistinguishably log in as a victim user at a genuine RP without the attacker learning the victim's secret credential (i.e., their password). This attack (along with a successful proof-of-concept) was disclosed to the Australian Signals Directorate on 19 August 2020, who passed the disclosure on to the ATO. The ATO indicated at a subsequent meeting in September 2020 that they had no intention of changing the protocol or otherwise mitigating the attack; as a result, the attack was publicly disclosed along with a warning to users of how to avoid it on 21 September 2020[56].

The attack takes advantage of the "feature" of the TDIF in which an IdP should not know the identity of the RP from whom an authentication request originates, to prevent the IdP from being able to track which RPs a user accesses. The inverted nature of the myGovID login process in which a user only ever enters their credentials into a trusted mobile app (acting as a CSP) violates the trust assumptions which internet users are taught to ensure they use online services securely; combined with the inability of the CSP to provide context for an authentication request (specifically, the RP from whom the request originated) to the user, it becomes very difficult for any users other than the most vigilant to detect this attack.

This attack serves as a compelling reinforcement to the issues caused by the TDIF's double-blind model discussed in chapter 4: the inability of the CSP or IdP to mitigate this kind of attack by informing the user where the login request originated is an inherent and unavoidable failing of double-blindness.

[56] A writeup of the attack is available at https://www.thinkingcybersecurity.com/DigitalID/, while a video demonstrating the attack is available at https://youtu.be/TgPdVbUbtBM

1 myGovID login process

When logging into a service using typical SSO, a user would normally be redirected to the IdP's website where they enter their credentials (in most cases, a username and password) and a two-factor authentication code from a trusted source (typically an app), before being redirected back to the RP. However, this is vulnerable to phishing attacks where a malicious RP "redirects" the user to a page controlled by the attacker which appears to an inattentive user to be the honest IdP, where the user will enter their credentials the user's credentials and thus reveal them to the attacker.

myGovID takes a different approach: instead of the user entering their credentials in the browser, where it is relatively easy for an attacker to present a phishing page and learn the user's secret credentials, a myGovID user instead only ever enters their secret credential into a trusted myGovID mobile app (the CSP). The steps to log in to a service using myGovID are as follows:

1. The user opens the website at which they can access a service operated by a TDIF-integrated RP.
2. The user clicks the "Login using myGovID" button (seen in Figure 7.1), initiating the authentication process. The appearance of this button is consistent across all RPs which are integrated with myGovID, so its presence is a strong signal of legitimacy to the user.
3. The RP redirects the user to the myGovID IdP via the IdX[57], where they are prompted to enter their email address.
4. These two steps occur simultaneously:
   (a) The IdP displays a 4-digit code to the user.
   (b) The myGovID mobile app (acting as the CSP) receives and displays a push notification indicating that an authentication request has been received by the IdP (seen in Figure 7.2).
5. The user enters their password into the myGovID mobile app.
6. The user verifies the authentication request in one of two possible ways[58]:
   - The user is presented with a code entry popup[59], where they enter the 4-digit code shown in their web browser by the IdP (seen in). This is, at the time of writing, the more common method.
   - The user is presented with a 4-digit code, which they must check

[57] The IdX is only visible to the user in this step as a browser redirect. According to the TDIF the IdX is required at this point to allow the user to select from the list of IdPs with which the IdX is integrated [06, FED-04-01-13], but this doesn't seem to occur in myGovID. Since the RP explicitly indicates which IdP is being used (which in itself goes against the double-blind property of the TDIF) and the IdX seems to only be integrated with a single IdP, it is perhaps unnecessary in practice; however, the TDIF does not explicitly make allowance for this situation.

[58] It's not clear how which method is presented to the user is determined. One appears to be the standard approach (entering the code), while the other appears to be the "future" approach (checking that the code matches on both the app and website). Presumably at some point only the future approach will be supported.

[59] The myGovID app does not allow screenshots of this screen to be taken, so no image is included.

matches the code shown in their web browser.

7. Assuming the user did not cancel, the request did not time out, and the code was successfully validated, the IdP redirects the user back to the RP via the IdX[60] along with the authentication response.
8. The user may now use the service.

Figure 7.1: The "Login with myGovID" button that indicates an RP is integrated with myGovID.

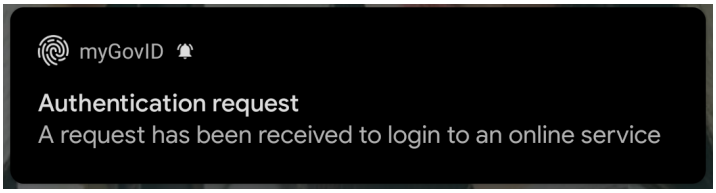

Figure 7.2: myGovID notification to prompt a user to verify an authentication request.

This reversed flow of information is a key factor in why this attack is unlikely to be detected by any but the most vigilant users, as it violates (or perhaps more accurately, sidesteps) the trust assumptions and mitigation techniques which users are taught regarding internet safety. The user never enters their password into their browser, which they are taught to be careful about; instead, they enter it into a trusted app installed from a trusted app store. This shifts the "dangerous" part of the login process away from the browser to a safer environment.

2 Attack scenario

This scenario involves two primary parties: Alice, a layperson who has used myGovID to log in to some authentic service in the past; and an attacker, who controls a service which we will call `nottrustworthy.com`. Alice believes that `nottrustworthy.com` is at least somewhat trustworthy and wishes to log in using myGovID, while the attacker wants to (fraudulently) log in as Alice at a service Alice uses, which we will call `alices-tax-service.gov.au`.

In this scenario, `nottrustworthy.com` and `alices-tax-service.gov.au` are both RPs, Alice is the user, the ATO provides the IdX, and myGovID is the IdP (and through its app, also the CSP). `nottrustworthy.com` is not necessarily an authentic RP integrated with myGovID (although it can be), but it

[60] Again, visible to the user only as a browser redirect. According to the TDIF the user should be required to consent to the release of attributes to the RP at this point [04, PRIV-03-09-01], but this does not seem to occur.

appears to users as if they are able to log in using myGovID by presenting a standard "Login with myGovID" button. `alices-tax-service.gov.au` is an authentic RP integrated with myGovID according to the TDIF specification.

As an established user of myGovID, we assume that Alice has the myGovID app installed and has used it before to log in to `alices-tax-service.gov.au`, and as such that she is familiar with the basics of how to use the app. We also assume that Alice is a layperson who isn't an expert in the app's trust assumptions. This attack does not require the app to be compromised in any way, and so Alice has no reason to distrust it—her trust is entirely valid.

3 The attack in detail

3.1 Setup

Before the attacker can successfully execute the attack, they must be able to present a convincing facade of authenticity to Alice. On the login page at `nottrustworthy.com` they must present a "Login with myGovID" button, which can be trivially copied from a genuine RP. Instead of redirecting a user to the IdX where they would perform a legitimate login, this button instead redirects the user to a page controlled by the attacker[61] which mimics the myGovID login page. While it is not trivial to perfectly mimic the legitimate page, it is not difficult to make one which is effectively indistinguishable to all but very attentive users.

While a diligent user who is familiar with the myGovID login process might realise that this page is not part of `https://mygovid.gov.au`, a layperson user such as Alice is unlikely to find anything suspicious about a page, identical to the one she is familiar with and presented by a service she believes to be integrated with myGovID, requesting her email address—this is a standard part of the login process, and nowhere does the myGovID website or app indicate that the only safe place to enter your email address is at `https://mygovid.gov.au`. It is important to note that this page does not at any point request a user's password—this is entered exclusively in the trusted app.

Finally, the attacker may present a genuine service at the end of the authentication process; this is outside the scope of the attack and primarily serves the purpose of protecting `nottrustworthy.com`'s seeming legitimacy.

With the login button and myGovID page in place, the attacker is ready.

3.2 Executing the attack

The attack begins when Alice accesses `nottrustworthy.com` and clicks the "Login with myGovID" button, and proceeds as follows:

[61] This could be part of `nottrustworthy.com` or elsewhere, depending on what the attacker judges to be more convincing.

1. Upon clicking the button, Alice is redirected to the attacker-controlled myGovID login page, which asks Alice to enter her email address. Alice enters her email address, giving the attacker knowledge of it.
2. Either manually or via an automated process[62], the attacker initiates a login process at `alices-tax-service.gov.au` and enters Alice's email address at the genuine myGovID login page.
3. The genuine myGovID login page displays a 4-digit code to the attacker, which is intended for Alice to validate using her myGovID mobile app.
4. The attacker presents the code to Alice[63] via the attacker-controlled myGovID login page in away that appears legitimate to Alice.
5. At the time the attacker initiated the login process at `alices-tax-service.gov.au`, Alice should have received a push notification from her myGovID app indicating an authentication process began[64]. Alice opens the app, enters her password, and validates the code using the technique described in section 1.
6. The attacker is now logged in at `alices-tax-service.gov.au` under Alice's identity. If the attacker wishes to protect their appearance of legitimacy, they may present a login success and genuine service to Alice at `nottrustworthy.com`.

3.3 Why the attack works

This attack succeeds specifically because of the TDIF's double-blindness: by preventing the IdP from knowing the identity of the RP, the IdP is also prevented from informing the user of to whom their data is being released. In this case, the push notification received by Alice's app is unable to inform her that the authentication request originated from `alices-tax-service.gov.au`, not from `nottrustworthy.com` like she expects. Alice therefore thinks that she is verifying her login attempt to `nottrustworthy.com`, but myGovID instead sees a request from `alices-tax-service.gov.au`, which is what is conveyed to Alice.

4 Impact of the attack

While this attack may appear superficially as if it is a standard phishing attack in which a user enters their secret credentials into an untrusted website controlled by the attacker, there are two key differentiating factors which make the attack different and of greater significance.

First, unlike a standard phishing attack, this attack does not aim to learn Alice's password. In most login processes, the user possesses two credentials: a

[62] In the proof-of-concept presented in the disclosure, this was manual via an attacker control panel; in a real attack, an automated process would be more effective.

[63] Again, this can be manual or automated; in the proof-of-concept attack, this is entered into the attacker control panel which automatically provides it to Alice's session.

[64] In my experience using myGovID, this notification sometimes arrives late (or even not at all). This inconsistent timing makes any delay beginning a login session at the target RP (and thus a delay in the notification's arrival) less noticeable.

non-secret credential which identifies them, such as a username or email address; and a secret credential, such as a password. The goal of a phishing attack is to harvest the victim's secret credential. However, in this attack Alice's password is never entered anywhere other than a trusted (and, indeed, trustworthy) app installed by Alice to a device in Alice's control, from a trusted source (the appropriate app store for Alice's phone). In fact, learning Alice's password offers no utility to the attacker at all—they can successfully and almost undetectably hijack Alice's login attempt without learning it, and the only way they could make use of it is if they had access to Alice's mobile device. Users are often warned about entering passwords into untrusted sites as part of a typical internet-use learning process; however, because of the reversed information flow in the myGovID login process this never occurs, and so the attack avoids perhaps the single most deeply ingrained piece of defensive knowledge users are taught. Users are taught much less about giving out their non-secret credential, so the non-`mygovid.gov.au` login page isn't inherently suspicious.

Second, the attack is actively facilitated and legitimised by a trustworthy component of the myGovID system. Even if Alice *is* suspicious, her suspicion is soon assuaged by the notification from her trusted app indicating that a genuine authentication request has been received by the myGovID IdP. In addition, the app will only accept a valid code matching an authentication request for the correct user—an attacker cannot just display an arbitrary value. If Alice enters the code displayed by the attacker and it is rejected, it will raise her suspicion; however, this defence mechanism fails to achieve anything in this attack, because the code displayed to Alice is for a genuine (as far as myGovID can tell) authentication request. Between the notification and subsequent acceptance of the code Alice is displayed, the most trustworthy participant in the login process not only fails to suggest that something might be wrong but actively indicates that Alice is participating in a genuine login.

This attack is detectable by a vigilant user who carefully checks where the login page is from and knows in advance that it is only safe if it is an HTTPS-protected URL on the domain `mygovid.gov.au`. However, most people are imperfect at this. While modern browsers do offer TLS-related defences such as indicating to a user when the page they are visiting is not TLS protected or uses an invalid certificate, this doesn't work in every situation—and assuming that `nottrustworthy.com` uses valid TLS, browsers would not flag anything wrong in this attack.

Expectations of user vigilance are further complicated by the myGovID login process being confusing and inconsistent even under normal circumstances. The app notification is often delayed from the login attempt, sometimes arriving after the time window for the code to be entered expires, or even not at all. If a user begins a login at an RP, cancels it, then begins it again, when they open the app the code entry popup they're presented with will be for the original login—it will reject the code the user is presented with with only a cryptic "Something is wrong with the code. Try again." error message and no indication of what is wrong. The user can only proceed by closing the code entry popup, after which

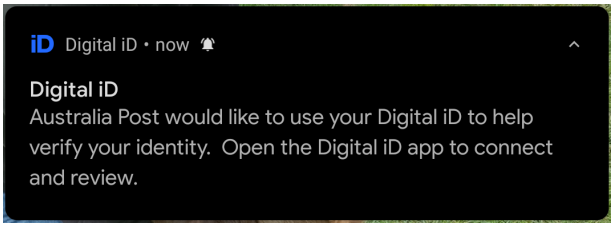

Figure 7.3: The notification raised by Digital iD when using it to authenticate identity for Australia Post's Mail Hold service. The originating RP is clearly identified.

they are presented with a second code entry popup which is indistinguishable from the first, except that this one will accept the code—at no point does the user have any reason to believe that they are doing anything wrong.

User experience which teaches users to expect and ignore inconsistencies undermines the guidelines for internet safety that they've learnt. The TDIF's insistence on double-blindness removes one of the easiest and best mitigations for this: showing the user the RP from whom the authentication request originated. This raises the question of whether to prioritise user privacy or to protect their security. While the TDIF's answer to this question is seemingly clear—user privacy is the favourite—the waters are muddied by the other IdP in the TDIF.

4.1 Comparison to Digital iD

Digital iD's approach to the login process is similar to myGovID. The user clicks a "Verify your identity with Digital iD" button which opens a window where they can enter an identifier[65]; the user enters the identifier associated with their mobile app, which triggers a push notification to be sent to the app; the user enters a passcode to unlock the app; and finally the user verifies the details of the authentication request. However, there is one clear and important difference: the user is told the identity of the originating RP, as can be seen in Figure 7.3[66].

While this clearly should not be allowed according to the double-blind property of the TDIF, Digital iD is nevertheless an accredited IdP[67]. This inconsistency between the behaviour of the only two accredited IdPs in the TDIF makes the answer to the previously raised question much less clear. The TDIF clearly indicates that it favours user privacy[68], and yet it allows and accredits violations of the main property it uses to enforce this privacy. This, of course, gives no answer to our question, but it does raise a new one: why does myGovID not do the same as Digital iD?

The obvious answer is that myGovID's IdX does not allow it to do so, since it does hide the identity of the originating RP in the authentication request

[65] Instead of an email address, Digital iD uses a phone number.
[66] It is also shown in the app, but the app disallows screenshots so no image is provided.
[67] This is discussed more in chapter 8.
[68] At least nominally.

from the IdX to the IdP. However, despite succeeding at hiding the identity of the RP in the OIDC request itself, the identity of the RP is still leaked by the IdX through the HTTP Referer header[69]. This means that despite the IdX's existence, the myGovID IdP still has knowledge of the RP's identity. Given that one IdP learns the identity of the RP by design and the other IdP learns it in spite of design, it is clear that the TDIF doesn't take enforcing double-blindness seriously[70]. Based on this, there is no compelling reason that myGovID could not follow the same route as Digital iD and communicate the RP's identity to the user.

5 Mitigations

Mitigations for this attack are limited within the scope of the TDIF as it currently exists. The most effective mitigation without requiring a redesign of the myGovID login process is to use the existing knowledge that the myGovID has of the RP's identity[71] to display it to the user in the myGovID app, as is done in Digital iD. This provides a simple defence mechanism for users, but does go against the TDIF's double-blind property. However, given the previously discussed acceptance of this behaviour in Digital iD, applying consistency would suggest that it should be allowed[72].

The ATO have expressed no inclination to implement a fix of this nature, or to address this attack at all. As a consequence, the onus for mitigating this attack falls on users. A user's best chance of mitigating the attack is to avoid using myGovID where possible. However, this is clearly not practical, as a growing number of government services exclusively use it as an authentication mechanism. In that case that a user must use myGovID, the best they can do to avoid this attack is to carefully check that the login page is protected by TLS and hosted at `https://mygovid.gov.au`. This is clearly not a practical solution—it requires both foreknowledge of the attack and technical knowledge about TLS. However, due to the disinclination of the ATO to address this attack, it is the best mitigation available.

6 Implications for the TDIF

The solution offered above yet again raises questions about the value of the double-blindness property. While the presence of an IdX does offer some benefits beyond that such as easier integration with the TDIF federation, it is clear that,

[69] This was reported to the ATO around the same time as this attack was disclosed. However, it does not appear to have been fixed as of 16 November 2020.

[70] Or, at least, that the accreditation process isn't sufficient in this regard. Either way, it fails to enforce double-blindness in practice.

[71] While modern browsers make it difficult to spoof the Referer header, it probably isn't a suitable mechanism to rely on for this purpose. If the ATO were to opt to take this approach, they should instead use a more formal mechanism.

[72] Consistency is not, however, something that the TDIF seems to prize highly.

on top of the privacy issues discussed previously in this thesis, it can endanger the security of TDIF users. Given that the "federation" is currently a federation only in name and users of myGovID-integrated government services cannot opt to use a different IdP like the TDIF intends[73], users are left with no effective defence mechanism against myGovID's security failings. Since double-blindness fails deliberately in Digital iD and accidentally in myGovID, they also do not gain its claimed privacy benefits.

[73] There is the future possibility of IdPs being added which could be used to log into government services. However, given that the next suggested addition is a separate IdX for the financial sector, it's unlikely that any new IdPs in that would also integrate with myGovID-compatible services, so the timeline for this happening is entirely unknown.

Chapter 8

Meta issues and implementation inconsistencies in the TDIF specification

The problems with the TDIF are not limited only to the design it presents and its implementation. There are a number of issues in the writing and presentation of the specification which make it either difficult or impossible for even an honest, diligent party to implement the part of the TDIF which is appropriate for its role. This chapter lays out a non-exhaustive list of some of these issues.

1 Issues arising from the TDIF v1.5 to TDIF Release 4 update

While it is difficult to ascribe a clear cause for many of these issues, a number of them are very likely caused by the format changes and document rearrangements which occurred in the most recent major TDIF update.

1.1 Removal of SHOULD requirements resulted in erroneous MAY requirements

One of the more significant changes made in TDIF Release 4 was the removal of **SHOULD** (and **SHOULD NOT**) requirements. In TDIF v1.5, a **SHOULD** requirement was a slightly weakened version of a **MUST** requirement:

> **SHOULD** – means an Applicant is required to meet this requirement unless there is a valid reason for them to ignore it. The Applicant must seek agreement from and provide evidence to the Trust Framework Accreditation Authority before it can ignore this requirement. Failure to meet this requirement will impact the Applicant's ability to achieve and maintain TDIF accreditation. (*TDIF Overview and Glossary*, §4.7)

This keyword was often used in requirements governing things such as token lifetimes:

> For clients using the Authorization Code grant type, access tokens **SHOULD** have a valid lifetime no greater than one hour and refresh tokens, if issued, **SHOULD** have a lifetime no longer than 24 hours. (*TDIF OpenID Connect 1.0 Profile*, §2.7.6)

Many of the same requirements still exist. It's unclear if there was a planned approach to how they were modified to fit the new version; the presumed approach is that such requirements would be either removed or recategorised as fully mandatory (*MUST*) or fully optional (*MAY*). A *MUST* requirement is an absolute requirement while a *MAY* requirement is optional:

> *MAY*. Means truly optional. This requirement has no impact on an Applicant's ability to achieve or maintain TDIF accreditation if it is implemented or ignored. Source: TDIF.
>
> *MUST*. Means an absolute requirement of the TDIF. Failure to meet this requirement will impact the Applicant's ability to achieve and maintain TDIF accreditation. Source: TDIF. (*01 - Glossary of Abbreviations and Terms*, §2)

There are two problems in how this update was carried out.

Meaningless requirements

It should be immediately clear that the decision of whether a **SHOULD** requirement should become a *MUST* or *MAY* is nuanced and should be decided on a case-by-case basis. However, this is not how the TDIF update was carried out. There are requirements in which the wording is exactly identical to the previous **SHOULD** version but which are now categorised as *MAY*. In some cases, this may be intentional. However, in others it is clearly not, as some of the new **SHOULD** versions make little to no sense, and are seemingly the result of direct find-and-replace operations.

Consider, for example, the following two requirements:

> **TDIF Req**: OIDC-02-07-16; **Updated**: Mar-20; **Applicability**: X For clients using the Authorization Code grant type, access tokens MAY have a valid lifetime no greater than one hour.
>
> **TDIF Req**: OIDC-02-07-17; **Updated**: Mar-20; **Applicability**: X Refresh tokens, if issued, MAY have a lifetime no longer than 24 hours. (*06B - OpenID Connect 1.0 Profile*, §2.7.3)

While the conventional use of the word "may" makes sense in this context ("may [be] no greater than" meaning "must be less than or equal to"), the TDIF reading of *MAY* means "can optionally". This clearly makes no sense: "access

tokens can optionally have a valid lifetime no greater than one hour" is the same as not having a rule about lifetimes at all. This is additionally compounded by use of the conventional meaning of "may" occurring in the same documents, which gives precedent for the same word to be read as the meaning which makes the rule make sense (but is incorrect).

This situation leads to ambiguity, which in a formal specification is not satisfactory. Because the reading that makes sense (the conventional meaning) is technically the invalid meaning (because it doesn't follow the strict TDIF definition of the keyword), it encourages parties implementing the specification to either ignore the specification in favour of common sense or to stick to the correct but clearly nonsensical rules the specification sets out. Neither of these are good options.

Scope of changes not limited to TDIF requirements

The TDIF itself references a number of other specifications which cover many topics, one of which is the the National Institute of Standards and Technology (NIST) special publication 800-63B (*Digital Identity Guidelines – Authentication and Lifecycle Management* [19]). This publication has its own definition of SHOULD requirements:

> The terms "SHOULD" and "SHOULD NOT" indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. (*Digital Identity Guidelines – Authentication and Lifecycle Management*)

This definition is clearly stronger than the definition of *MAY* in the TDIF. However, rather than using the definition from the NIST publication to preserve its intent and strength of requirements, the TDIF instead opts to weaken all NIST SHOULD requirements to TDIF *MAY* requirements:

> 'SHOULD' statements in NIST should also be read as 'MAY' statements in the TDIF. (*05 - Role Requirements*, §4)

In the case that this change had been made with due consideration to the affected requirements, it would be reasonable. However, given that it is clear that the change from **SHOULD** to *MAY* was not even considered thoroughly for the TDIF's own requirements, it seems exceedingly unlikely that such consideration was given to external requirements.

1.2 Unclear ends of requirements

TDIF Release 4 restructured requirements to make the separation between different requirements more clear and to allow them to be uniquely referenced.

Previously, what was a requirement was indicated only by the use of specific keywords, but in the new release requirements are named and, in general, cover a single distinct point. While this makes it more clear where a requirement begins and ends and divides them into clear pieces, the authors failed to update every requirement correctly. As such, there are a number of places where requirements seem to flow into each other.

Consider, for example, the following contiguous group of requirements, and in particular requirement OIDC-03-08-08:

> **TDIF Req**: OIDC-03-08-07; **Updated**: Mar-20; **Applicability**: I
>
> When the ACR values are marked as an essential claim, the Identity Provider MUST return a value that matches the requested values.
>
> **TDIF Req**: OIDC-03-08-08; **Updated**: Mar-20; **Applicability**: I
>
> If the End-User is unable to achieve a level of assurance that matches at least one of the ACR values requested by an Exchange then an authentication error response *MUST* be returned.
>
> Where the `acr_values` parameter is used a space separated set of ACR strings must be provided. The Authentication Context Class satisfied by the authentication performed is returned as the `acr` claim Value.
>
> ```
> acr_values=urn:id.gov.au:tdif:acr:ip3:cl2
> urn:id.gov.au:tdif:acr:ip3:cl3 urn:id.gov.au:tdif:acr:ip4:
>     cl3
> ```
>
> When requesting the `acr` claim using this parameter it is requested as a voluntary claim i.e. cannot be marked as essential.
>
> **TDIF Req**: OIDC-03-08-09; **Updated**: Mar-20; **Applicability**: I
>
> When the acr claim is not marked as essential, i.e. they are a voluntary claim, the Identity Provider *MAY* return the level of assurance that the End-User was able to achieve.
>
> The specification of the `acr` claim within the request object is the preferred method for requesting the `ACR`.
>
> **TDIF Req**: OIDC-03-08-10; **Updated**: Mar-20; **Applicability**: X
>
> The Identity Exchange MUST determine if the returned ACR meets the minimum requirement for the authentication context that was requested.
>
> (*06B - OpenID Connect 1.0 Profile*, §3.8.3)

The first sentence of requirement OIDC-03-08-08 is a genuine requirement which specifies when error responses must be returned to an IdX from an IdP when a user is not authenticated to an appropriate level of assurance. However, the second part of the requirement is seemingly about an entirely different topic (the format of the `acr_values` request parameter) while the last sentence is a related but different topic again.

Due to the way requirements are typeset in the TDIF specification, it's not clear if OIDC-03-08-08 is intended to be read as a single requirement encompassing all three seemingly unrelated sections, or if only the first part is the requirement and the following sections (until the beginning of the next requirement) are separate informative sections, particularly given that they cover a different topic to the requirement proper. Requirement OIDC-03-08-09 is also similar, but its second section *is* related to the main section so it would be reasonable for it to be a single requirement.

This makes it difficult for a potential applicant to understand exactly which parts of the specification are requirements which they are required to implement and which parts are merely informative. This is not a difficult problem to solve, requiring only some simple change to the typesetting of requirements which clearly delineates the end of a requirement, such as placing a requirement inside a box.

2 References to invalid or non-existent sections

TDIF documents frequently make references to specific sections of other documents to reduce duplication. However, in a number of cases the referenced section doesn't actually exist. For example, the TDIF includes the following reference:

> Relying Parties approved to request Verified documents as restricted attributes under section 3.6.1 of the TDIF: 05 Role Requirements. (*06D - Attribute Profile*, §2.3)

While the document being referenced does contain a section 3.6, that particular sections contains no subsections, so section 3.6.1 doesn't exist. It's difficult to determine exactly where the correct target of the reference is: even though section 3.6 is about attributes, it contains no mention of relying parties, and neither section 3.6 nor the following section 3.7[74] contain any mention of "restricted attributes".

There is also at least one case of a reference to the wrong document entirely: section 2.6.1 of the OIDC profile [06B] indicates that additional OIDC scope values are described in *06A - Federation Onboarding Guidance*. However, the indicated document contains no such descriptions, which instead appear in *06D - Attribute Profile*.

While I did not collect an exhaustive list of places at which invalid references exist, this is not the only place it occurs[75]. These references, as with many of the other issues in this chapter, make it much more difficult for a potential TDIF applicant to navigate the TDIF specification.

[74] Which is about attribute disclosure, so seems likely to be a potential target.

[75] For example, *06A - Federation Onboarding Guidance* contains a reference to the non-existent section 2.2.1.1 in *06 - Federation Onboarding Requirements*; presumably the reference should be to section 2.3.1.1 instead, which sets out the requirements relevant to the topic being discussed.

3 Unclear use of requirement keywords

Despite the TDIF clearly defining the keywords which identify a requirement, there are cases where either a keyword is used in a place such that something has become a requirement that seems unlikely to be intended, or a keyword has been used informally (i.e., without the specific formatting) in a requirement where it seems likely that it should have been used formally.

For example, consider the following requirement:

> **TDIF Req**: OIDC-03-07-06; **Updated**: Mar-20; **Applicability**: I Endpoints and parameters specified in the Discovery document *MAY* be considered public information regardless of the existence of the discovery document. (*06B - OpenID Connect 1.0 Profile*, §3.7.2.4)

It seems unlikely that this is intended to be a requirement: it requires nothing to be done by an applicant regardless of whether they choose to "implement" it or not, and it is clearly unverifiable. Instead, it seems more likely that it should be considered an informative note to an applicant of how something may be used, in which case it should not use the *MAY* keyword and it should not be a labelled requirement.

As an example of the opposite case, consider again requirement OIDC-03-08-08 shown in subsection 1.2. The first sentence is a genuine requirement and uses a requirement keyword. The second section uses the keyword "must" but not formatted according to the definition of requirement keywords. However, given that the second section is clearly about an unrelated topic to the first section, it seems likely that it was intended to be a distinct requirement from the first sentence but was not noticed due to not using the appropriately formatted keyword.

This confusing use of requirement keywords makes it difficult to judge the exact intent of the TDIF. There are many cases where a requirement keyword is used in a section which reads like a requirement and would make sense to be a requirement but is not labelled or typeset as one, such as in the following:

> All clients must validate the signature of an ID Token before accepting it using the public key of the issuing server, published in JSON Web Key (JWK) format. (*06B - OpenID Connect 1.0 Profile*, §2.6.3.1)

It's unclear whether this was indeed intended to be a requirement[76] or not, and a well-meaning but naive TDIF applicant would be entirely correct to ignore it, as anything which is not formatted as a requirement according to the TDIF's strict formatting is not clearly required. If we assume that situations like these *are* intended to be requirements, not having them formatted as such clearly weakens the TDIF by removing important requirements. Even if they are *not* intended to be requirements, they still negatively impact the usefulness of the TDIF by making it more difficult for applicants to understand what is expected of them.

[76] Even if it wasn't intended to be one, it certainly *should* be one.

4 Implementation inconsistencies

The following issues are present in the existing implementations of the TDIF, but are inconsistent with its design and requirements.

4.1 RPs explicitly indicate which IdP is going to be used

Because the TDIF currently only includes two IdPs which service entirely disjoint sets of RPs, RPs indicate to the user which IdP will be used[77]. This is entirely incongruous with the TDIF's design, in which a major purpose of the IdX's existence is to hide the identity of the IdP chosen by the user from the RP and vice versa.

In theory, the RP should never be able to suggest an IdP to be used[78] due to the requirement that the IdX must allow the user to select which one they want to use [06, FED-04-01-13]. However, because the TDIF only includes two IdPs, only one of them uses an IdX (so there is never a case where an IdX could allow a user to select from more than one IdP), and the two IdPs do not share any overlapping RPs, there is never a choice available.

Additionally, there is no provision in FED-04-01-13 which allows an IdX to avoid showing the user a list of IdPs to choose from[79]. Despite this, neither myGovID nor Digital iD allow users to select an IdP during the authentication process. This is a clear violation of the requirement which presumably should have been addressed during the accreditation process.

4.2 No IdX dashboard for reviewing consent in myGovID

The TDIF includes provision for a *user dashboard* at an IdX, through which a user can access their historical interactions and consent associated with a specific identity at an IdP [06A, §4.1.2]. This dashboard allows the user to view both requested and returned attributes[80] and to revoke ongoing consent which they have given in the past. Providing this dashboard is mandatory, as specified in requirement [06, FED-04-01-06].

However, such a dashboard does not seem to exist for myGovID. There is no mention of the dashboard on the myGovID website, nor is there any point in the authentication process mentioning or providing access to such a dashboard. It

[77] RPs use a "Login with IdP" button specific to the IdP they're integrated with, as described in chapter 7, so it is immediately and unambiguously clear to a user which IdP they are using.

[78] There are cases where only a single IdP integrated with the IdX may be able to service the specific requirements of the authentication request, but even then it shouldn't be possible for the RP to specifically request that IdP—particularly if there is a possibility of new IdPs being added, which may then be able to service the request.

[79] Requirement [06, FED-04-01-14] allows the IdX to filter the list to only IdPs capable of servicing the request, but nowhere do either of the requirements indicate that the IdX can skip the selection process if the list of IdPs contains only a single entry.

[80] It does not store the *values* of the attributes, but rather only which attributes were requested and returned.

is perhaps possible that such a dashboard exists but is not publicly known, but even if that were the case it does not satisfy the intent of the explicit requirement that such a dashboard is provided. Not providing a user dashboard is a clear violation of an unambiguous requirement, which again should have been found and addressed during the accreditation process.

Digital iD does provide an activity log in its mobile app; however, because I did not provide Digital iD with any personal identity documents I was unable to use it to authenticate anywhere and therefore could not investigate how the activity log functions. Requirement FED-04-01-06 does indicate that the dashboard should be provided by the IdX, but since Digital iD does not seem to use an IdX the app is arguably the most reasonable place.

Chapter 9

Conclusion

In this thesis I have presented a number of issues in the TDIF specification, affecting every part of the TDIF from the way it is written to its architecture to its implementation. The TDIF purports to be a "privacy-preserving" model, and yet it features a central entity which mediates almost all interactions and has specification-sanctioned power to perform surveillance of user activities to a greater level than the activity tracking it aims to prevent. The specification features inappropriate use of private identity document attributes as sources of cryptographic entropy and is riddled with clarity and correctness issues which impede understanding. The accreditation process intended to enforce that participants in the TDIF implement the requirements appropriate to their role fails to enforce even the most fundamental requirement—the use of an identity exchange. Practical attacks exist against real implementations of the TDIF which are facilitated by its double-blindness property. And in the implementation that exists, the identity federation does not offer users any choice of identity provider, and nor does it achieve double-blindness.

These problems are likely only scratching the surface, given the limited time and breadth-over-depth focus of this work. However, they permeate almost every aspect of the TDIF's design, which raises deeply concerning questions about how much care and attention went into the design process of a framework which aims to be so widely used and handles personal identity data. While there are indeed clear benefits to the use of a brokered identity model such as simple onboarding and easy separation of identity sectors, these benefits do not outweigh the myriad concrete issues and privacy concerns which have been raised in this work.

The TDIF is a government system which has been under development for several years—there is no justification for the depth and breadth of observed issues in its design. Users may have no recourse if they wish to avoid its use due to privacy concerns, as many myGovID-integrated RPs offer no alternative authentication method—which defies the TDIF's opt-in guiding principle. The Australian government has a responsibility to immediately invest significant effort into fixing the TDIF, and the issues presented in this thesis offer a number of places to start.

1 Where to from here?

1.1 Recommendations for the TDIF

The TDIF as currently designed and implemented does not meet its own guiding principles—it is not immediately obvious that a brokered model without technical means to preserve privacy even *can* meet them. We recommend a careful re-evaluation of the priorities of the TDIF, and a consideration of other options which may meet its goals. In particular we recommend the DTA investigates the following alternatives:

Use of a public key infrastructure-based system. A PKI-based system such as those used widely in the European Union is a promising alternative to the brokered model. PKI-based digital identity management is widespread and well understood, with published standards for international interoperability. It offers many of the security and privacy benefits that the TDIF aims to have, but with the added advantage that there is no entity who can meaningfully track user activity, as authentication occurs without the direct involvement of a central authority. As such, we believe this to be the most promising candidate to take the place of the TDIF.

Use of a simpler, pairwise model instead of a complex brokered model with poor privacy and security properties. We recognise that implementing a PKI-based authentication system on a country-wide scale is a daunting task. As a simpler holdover solution while such a system is developed, vastly reducing the complexity of the TDIF to a one-legged OIDC-based system offers better privacy and security than the current version. While the obvious objection to this is the ability of an IdP to track user activity, we contend that an IdX in the TDIF has significantly more power to do the same, and so this is not a compelling argument. Nevertheless, while we believe one-legged OIDC to be an improvement over the current TDIF, we believe PKI-based authentication to be a better potential solution.

1.2 Future research

This thesis raises many questions and points out many problems, but it does not offer many solutions. There is much room for future research into ways to mitigate the issues described this work, particularly in novel ways to improve the security and privacy of a brokered identity model—the potential solutions proposed by Brandão et al. [4] offer a clear starting point, but are certainly not the only approaches that could be taken.

This work also explored the TDIF widely but not in-depth. As such, there is much of the TDIF which has not yet been analysed in detail—publicly available research into the security of such a large government framework offers clear value

to the Australian public and may lead to concrete improvements in the TDIF's design.

Bibliography

[1] Aichholzer, Georg and Strauß, Stefan, "Electronic identity management in e-Government 2.0: Exploring a system innovation exemplified by Austria", *Information Polity* 15.1, 2 (2010), pp. 139–152.

[2] Arora, Siddhartha, "National e-ID card schemes: A European overview", *Information Security Technical Report* 13.2 (2008), pp. 46–53.

[3] Boeyen, Sharon et al., "Trust Models Guidelines", *Draft. OASIS* (2004).

[4] Brandão, Luís TAN, Christin, Nicolas, Danezis, George, et al., "Toward mending two nation-scale brokered identification systems", *Proceedings on Privacy Enhancing Technologies* 2015.2 (2015), pp. 135–155.

[5] Camenisch, Jan and Van Herreweghen, Els, "Design and Implementation of the Idemix Anonymous Credential System", *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, Washington, DC, USA: Association for Computing Machinery, 2002, pp. 21–30, ISBN: 1581136129, DOI: 10.1145/586110.586114.

[6] Cantor, Scott et al., *Assertions and Protocols for the OASIS Assertion Markup Language (SAML) V2.0*, tech. rep., Organization for the Advancement of Structured Information Standards (OASIS), 2005, URL: https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[7] Cap, Clemens H. and Maibaum, Nico, "Digital Identity and its Implication for Electronic Government", *Towards the E-Society: E-Commerce, E-Business, and E-Government*, ed. by Beat Schmid, Katarina Stanoevska-Slabeva, and Volker Tschammer, IFIP International Federation for Information Processing, Boston, MA: Springer US, 2001, pp. 803–816, ISBN: 978-0-306-47009-7, DOI: 10.1007/0-306-47009-8_59, URL: https://doi.org/10.1007/0-306-47009-8_59 (visited on 10/06/2020).

[8] D. Hardt, Ed., *The OAuth 2.0 Authorization Framework*, RFC 6749, IETF, Oct. 2012, URL: https://tools.ietf.org/html/rfc6749.

[9] Dhamija, R. and Dusseault, L., "The Seven Flaws of Identity Management: Usability and Security Challenges", *IEEE Security Privacy* 6.2 (Mar. 2008), pp. 24–29, ISSN: 1540-7993, DOI: 10.1109/MSP.2008.49.

[01] Digital Transformation Agency, *01 - Glossary of Abbreviations and Terms*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/TDIF%2001%20Glossary%20-%20Release%204%20V1.1.pdf.

[02] Digital Transformation Agency, *02 - Overview*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2002%20Overview%20-%20Release%204%20Final.pdf.

[03] Digital Transformation Agency, *03 - Accreditation Process*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2003%20Accreditation%20Process%20-%20Release%204%20Final.pdf.

[04] Digital Transformation Agency, *04 - Functional Requirements*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2004%20Functional%20Requirements%20-%20Release%204%20Final.pdf.

[04A] Digital Transformation Agency, *04A - Functional Guidance*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2004A%20Functional%20Guidance%20-%20Release%204%20Final.pdf.

[05] Digital Transformation Agency, *05 - Role Requirements*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2005%20Role%20Requirements%20-%20Release%204%20Final.pdf.

[05A] Digital Transformation Agency, *05A - Role Guidance*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/

s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2005A%20Role%20Guidance%20-%20Release%204%20Final.pdf.

[06] Digital Transformation Agency, *06 - Federation Onboarding Requirements*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2006%20Federation%20Onboarding%20Requirements%20-%20Release%204%20Final.pdf.

[06A] Digital Transformation Agency, *06A - Federation Onboarding Guidance*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2006A%20Federation%20Onboarding%20Guidance%20-%20Release%204%20Final.pdf.

[06B] Digital Transformation Agency, *06B - OpenID Connect 1.0 Profile*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2006B%20OpenID%20Connect%201.0%20Profile%20-%20Release%204%20Final.pdf.

[06C] Digital Transformation Agency, *06C - SAML 2.0 Profile*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2006C%20SAML%202.0%20Profile%20-%20Release%204%20Final.pdf.

[06D] Digital Transformation Agency, *06D - Attribute Profile*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2006D%20Attribute%20Profile%20-%20Release%204%20Final.pdf.

[07] Digital Transformation Agency, *07 - Annual Assessment*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%2007%20Annual%20Assessment%20-%20Release%204%20Final.pdf.

[10] Digital Transformation Agency, *Stakeholder and community feedback*, Trusted Digital Identity Framework Release 4, Commonwealth of Australia (Digital Transformation Agency), 2020, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-framework-4-final/TDIF%20-%20Stakeholder%20and%20Community%20Feedback%20-%20Release%204%20Final.pdf.

[11] Digital Transformation Agency, *Stakeholder and community feedback (component 3)*, Trusted Digital Identity Framework Document, Commonwealth of Australia (Digital Transformation Agency), 2019, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-component-3-stakeholder-community-feedback-summary.pdf.

[12] Digital Transformation Agency, *TDIF Architecture Overview*, Trusted Digital Identity Framework Document, Commonwealth of Australia (Digital Transformation Agency), 2019, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-architecture-overview.pdf.

[13] Digital Transformation Agency, *TDIF OpenID Connect 1.0 Profile*, Trusted Digital Identity Framework Document, Commonwealth of Australia (Digital Transformation Agency), 2019, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-openid-connect-1-0-profile.pdf.

[14] Digital Transformation Agency, *TDIF Overview and Glossary*, Trusted Digital Identity Framework Document, Commonwealth of Australia (Digital Transformation Agency), 2019, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/tdif-overview-glossary.pdf.

[15] Fett, D., Küsters, R., and Schmitz, G., "The Web SSO Standard OpenID Connect: In-depth Formal Security Analysis and Security Guidelines", *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, Aug. 2017, pp. 189–202, DOI: 10.1109/CSF.2017.20.

[16] Fett, Daniel, Küsters, Ralf, and Schmitz, Guido, "A Comprehensive Formal Security Analysis of OAuth 2.0", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16: 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria: ACM, Oct. 24, 2016, pp. 1204–1215, ISBN: 978-1-4503-4139-4, DOI: 10.1145/2976749.2978385, URL: https://dl.acm.org/doi/10.1145/2976749.2978385 (visited on 05/10/2020).

[17] Fioravanti, Fabio and Nardelli, Enrico, "Identity management for e-government services", *Digital government*, Springer, 2008, pp. 331–352.

[18] Grassi, P and Varley, M, *International Government Assurance Profile (iGov) for OpenID Connect 1.0 - Draft 02*, tech. rep., 2017, URL: https://openid.net/specs/openid-igov-openid-connect-1_0-02.html.

[19] Grassi, Paul A, Garcia, Michael E, and Fenton, James L, *Digital Identity Guidelines – Authentication and Lifecycle Management*, NIST Special Publication 800-63B, 2017, URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf.

[20] Hanson, Fergus, *Preventing another Australia Card fail: Unlocking the potential of digital identity*, 2018, URL: https://www.aspi.org.au/report/preventing-another-australia-card-fail (visited on 09/01/2019).

[21] J. Bradley, Ed., Lodderstedt, T., and Zandbelt, H., *Encoding claims in the OAuth 2 state parameter using a JWT*, Internet-Draft draft-bradley-oauth-jwt-encoded-state-08, IETF, Jan. 2018, URL: https://tools.ietf.org/id/draft-bradley-oauth-jwt-encoded-state-08.html.

[22] Jones, M, *JSON Web Key (JWK)*, RFC 7517, IETF, May 2015, URL: https://tools.ietf.org/html/rfc7517.

[23] Jones, M and Hildebrand, J, *JSON Web Encryption (JWE)*, RFC 7516, IETF, May 2015, URL: https://tools.ietf.org/html/rfc7516.

[24] Jones, M. et al., *OAuth 2.0 Token Binding*, Internet-Draft draft-ietf-oauth-token-binding-08, IETF, Oct. 2018, URL: https://tools.ietf.org/html/draft-ietf-oauth-token-binding-08.

[25] Li, Wanpeng, Mitchell, Chris J, and Chen, Thomas, "Mitigating CSRF attacks on OAuth 2.0 Systems", *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018 16th Annual Conference on Privacy, Security and Trust (PST), Aug. 2018, pp. 1–5, DOI: 10.1109/PST.2018.8514180.

[26] Mainka, C. et al., "SoK: Single Sign-On Security — An Evaluation of OpenID Connect", *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, Apr. 2017, pp. 251–266, DOI: 10.1109/EuroSP.2017.32.

[27] Makaay, Esther, Smedinghoff, Tom, and Thibeau, Don, *Trust Frameworks for Identity Systems*, White Paper, Open Identity Exchange, July 2017, URL: https://openidentityexchange.org/networks/87/item.html?id=175.

[28] Maler, E. and Reed, D., "The Venn of Identity: Options and Issues in Federated Identity Management", *IEEE Security Privacy* 6.2 (Mar. 2008), pp. 16–23, ISSN: 1540-7993, DOI: 10.1109/MSP.2008.50.

[29] McKenzie, R., Crompton, M., and Wallis, C., "Use Cases for Identity Management in E-Government", *IEEE Security Privacy* 6.2 (Mar. 2008), pp. 51–57, ISSN: 1540-7993, DOI: 10.1109/MSP.2008.51.

[30] Murray, David et al., *Financial system inquiry*, Department of the Treasury (Australia), 2014.

[31] Parsovs, Arnis, "Estonian Electronic Identity Card: Security Flaws in Key Management", *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020, pp. 1785–1802, ISBN: 978-1-939133-17-5, URL: https://www.usenix.org/conference/usenixsecurity20/presentation/parsovs.

[32] Pfitzmann, Birgit and Waidner, Michael, "Federated Identity-Management Protocols", *Security Protocols*, ed. by Bruce Christianson et al., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 153–174, ISBN: 978-3-540-31836-1.

[33] Poller, Andreas et al., "Electronic identity cards for user authentication—promise and practice", *IEEE Security & Privacy Magazine* 10.1 (2012), pp. 46–54.

[34] "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", *OJ* L 257 (Aug. 2014), pp. 73–114.

[35] Richer, J, Grassi, P, and Varley, M, *International Government Assurance Profile (iGov) for OAuth 2.0 - Draft 02*, tech. rep., 2017, URL: https://openid.net/specs/openid-igov-oauth2-1_0-02.html.

[36] Sakimura, N, Bradley, J, and Jones, M, *JSON Web Signature (JWS)*, RFC 7515, IETF, May 2015, URL: https://tools.ietf.org/html/rfc7515.

[37] Sakimura, N, Bradley, J, and Jones, M, *JSON Web Token (JWT)*, RFC 7519, IETF, May 2015, URL: https://tools.ietf.org/html/rfc7519.

[38] Sakimura, N, Bradley, J, and Jones, M, *OAuth 2.0 Mix-Up Mitigation*, Internet-Draft draft-ietf-oauth-mix-up-mitigation-01, IETF, July 2016, URL: https://tools.ietf.org/html/draft-ietf-oauth-mix-up-mitigation-01.

[39] Sakimura, N et al., *OpenID Connect Core 1.0*, tech. rep., 2014, URL: https://openid.net/specs/openid-connect-core-1_0.html.

[40] Sakimura, N et al., *OpenID Connect Discovery 1.0*, tech. rep., 2014, URL: https://openid.net/specs/openid-connect-discovery-1_0.html.

[41] Scarfone, Karen, Jansen, Wayne, and Tracy, Miles, *Guide to General Server Security*, tech. rep. s 123, National Institute of Standards and Technology, 2008, URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-123.pdf.

[42] Shim, S. S. Y., Bhalla, Geetanjali, and Pendyala, Vishnu, "Federated identity management", *Computer* 38.12 (Dec. 2005), pp. 120–122, ISSN: 0018-9162, DOI: 10.1109/MC.2005.408.

Appendix A

This appendix outlines an unverified attack on myGovID based on the loose and conflicting validation requirements laid out in the TDIF OpenID Connect 1.0 Profile [06B]. No proof-of-concept was developed to verify this attack; however, regardless of the attack's success or lack thereof, the situation which leads to its potential existence reinforces further the issues identified in the rest of this thesis regarding the careless writing of the TDIF specification.

1 Potential validation attack in myGovID

OpenID Connect uses the parameters `state` and `nonce` to protect against cross-site request forgery (CSRF) and replay attacks, and the JSON Web Tokens (JWTs) it uses contain a claim `jti`, which uniquely identifies the JWT to prevent token reuse. Acceptable values and validation of these parameters are specified in the iGov OpenID Connect Profile [18] and iGov OAuth 2.0 Profile [35] on which the TDIF OpenID Connect Profile is based:

> Full clients and browser-embedded clients making a request to the authorization endpoint MUST use an unpredictable value for the state parameter with at least 128 bits of entropy. Clients MUST validate the value of the `state` parameter upon return to the redirect URI and MUST ensure that the state value is securely tied to the user's current session (e.g., by relating the state value to a session identifier issued by the client software to the browser). (Section 2.1.1 [35])

> jti: A unique JWT Token ID value with at least 128 bits of entropy. This value MUST NOT be re-used in another token. Clients MUST check for reuse of jti values and reject all tokens issued with duplicate jti values. (Section 3.2.1 [35])

> nonce: REQUIRED. Unguessable random string generated by the client, used to protect against CSRF attacks. Must contain a sufficient amount of entropy to avoid guessing. Returned to the client in the ID Token. (Section 2.1 [18])

Neither of the iGov profiles contain any mention of validating the value of `nonce`; however, validation is mentioned in the OpenID Connect specification [39] itself:

> If present in the ID Token, Clients MUST verify that the `nonce` Claim Value is equal to the value of the `nonce` parameter sent in the Authentication Request. If present in the Authentication Request, Authorization Servers MUST include a `nonce` Claim in the ID Token with the Claim Value being the nonce value sent in the Authentication Request. Authorization Servers SHOULD perform no other processing on `nonce` values used. (Section 2 [39])

If an RP uses and validates these parameters in the intended manner, then it should be safe from the attack described here. However, the TDIF introduces ambiguity around the validation of these parameters or deviates from the base specifications, and it does so in such a way that a naive RP implementing the TDIF as written risks being vulnerable to a simple replay attack in which the victim's ID token may be substituted into an ongoing authentication process initiated by the attacker. If the RP has not implemented the validation process correctly, this grants the attacker a login session under the victim's credentials.

This attack is not generalisable to every RP in the TDIF, however, as the TDIF specification dictates that the OIDC Authorization Code Flow is used:

> These clients *MUST* use the authorization code flow of OAuth 2.0 by sending the resource owner to the authorisation endpoint to obtain authorisation. (OIDC-02-01-02 and OIDC-02-01-02)

In the Authorization Code Flow, the RP requests an authorization code from the IdX's authorization endpoint. This code is then redeemed by the RP via a back-channel request (i.e., a request which goes directly from the RP to the IdX without involving the User Agent) for an ID token and an access token. This prevents an attacker from substituting a different ID token, which would render a conforming client immune to this attack.

However, myGovID RPs do not use the Authorization Code Flow as the TDIF specifies that they should; instead, they use the Implicit Flow. In the Implicit Flow, the ID and access tokens are returned directly from the IdX's authorization endpoint, which means they are sent to the RP via the User Agent. This allows an attacker to interrupt and modify the normal flow of requests to substitute in the victim's ID token.

1.1 Validation weaknesses in the TDIF

This attack relies on the following issues in the TDIF's OpenID Connect Profile:

1. While the TDIF maintains the validation requirements for the `state` parameter word-for-word from the iGov OAuth Profile (OIDC-02-06-02), it weakens the requirement that it uses an unpredictable value:

> Clients making a request to the authorisation endpoint *MAY* use an unpredictable value for the state parameter with at least 128 bits of entropy. This is recommended, but it is up to the Relying Party to decide what level of entropy is required in the state parameter. (OIDC-02-06-01)

As "*MAY*" in the TDIF means "entirely optional", this leaves room for a naive RP to correctly implement the state parameter by using a fixed value[81]. This requirement could potentially (and arguably equally validly) be interpreted as applying only to the *level* of entropy, not the requirement for entropy at all; however, as is often the case in the TDIF specification, it's entirely ambiguous which is the correct and intended reading.

2. The TDIF retains the requirement that a given `jti` value may not be reused for multiple tokens and its required entropy. However, it comes with two problems:
   (a) The first requirement is only provided when considering the identity exchange's role as the OP. The `jti` is not discussed at all when considering the role of the RP.
   (b) The requirement that the Client should check for `jti` re-use is not present anywhere in the TDIF itself.
3. As in the iGov Profile, `nonce` claim validation is not present in the TDIF and relies on the OpenID Connect specification.
4. The TDIF specification does include a requirement on token validation at the RP (OIDC-02-06-10). However, it only covers the claims `iss`, `aud`, `exp`, `iat`, and `nbf`. This introduces additional ambiguity: while the TDIF Profile does explicitly say that readers should use the base OpenID Connect specification to fill any gaps in implementation details in the TDIF Profile, the presence of an explicit requirement of what claims in the ID token must be validated and how suggests that token validation is not a gap, and thus a naive reader may consider those requirements to be complete.
5. While some components of the TDIF undergo accreditation to ensure that they have implemented the requirements correctly, relying parties are not accredited. This means that as long as an RP implements the technical components of the specification to the written standard it can consider itself correct and compliant, and no additional safeguards are included as part of the TDIF.

1.2 Preconditions of the attack

This attack has three main preconditions.

1. The RP uses the Implicit Flow instead of the Authorization Code Flow, which is the case for RPs integrating with myGovID.
2. The RP has implemented token validation to the minimum compliant extent specified in the TDIF OpenID Connect Profile, and it has implemented the `state` parameter to the minimum requirements (i.e., it has opted to

[81] Which, as I discuss later, is the case.

use a fixed or predictable value).

3. The attacker can obtain a valid ID token of a victim through some means. This might be possible using a CSRF attack (particularly due to myGovID's use of a predictable state value), but for the purposes of describing the basic attack, I will not provide a specific means of obtaining the token.

1.3 The attack in detail

1. The attacker begins a standard authentication process at the target RP (e.g., Login using myGovID).
2. The attacker allows the authentication process to proceed as normal, recording the state value and return URL observed.
3. The attacker pauses the authentication process at some point after control has been passed on to the IdX (such as when logging in at the IdP).
4. The attacker manually completes the authentication process by making a request to the return URL (as appropriate to the return method specified by the RP when it began the process) which provides the expected values as specified in Section 3.2.2.5 [39], substituting the victim's ID token instead of their own.
    - The target RP will then attempt to validate the token. The state value used is valid, belonging to the legitimate authentication process started by the attacker, and under precondition 1, the RP does not implement `jti` or `nonce` validation, as neither are specified in the TDIF. Thus, even though the victim's token has an invalid `nonce` value and a re-used `jti` value, it should be considered a valid token.
5. If the RP accepts the request, the attacker should be logged in under the victim's identity at the target RP.

1.4 Limitations

It is difficult to verify whether an RP integrated with myGovID is vulnerable to this attack. Because this attack is against a live government system, I erred on the side of caution and did not extensively test it against different RPs. Many of the available RPs also require a user to have verified themselves to at least IP Level 2, and further check that a user has the right authorisations before returning to the RP (requiring the attacker has authorisation at that RP, not just the victim).