CDR Data Standards security analysis

Chris Culnane
chris@castellate.com

Ben Frengley
ben.frengley@gmail.com

Vanessa Teague
vanessa@thinkingcybersecurity.com

June 24, 2022

Abstract

Contents

1 Introduction

The Consumer Data Right (CDR) Standards (henceforth the Data Standards) use an open and transparent development process, much more akin to normal international standards processes than to the closed processes more common within the Australian Commonwealth Government. It is important to emphasise the tremendous benefits of an open process, in which not only the standards themselves, but also all the decision making and prior input, are made public. The online GitHub repository[1] allows for active participation by the wider community, and allows new participants like us to understand the history of decisions and the ways in which prior concerns have been addressed. This allows for a process of ongoing improvement that is crucially important for security standards in an ever-changing world.

There is absolutely no reason that all Australian public-sector IT processes could not be equally open, both in the sense of publishing their findings and reports, and also in inviting public contributions from the wider community.[2] They would produce better results if they were.

1.1 This report

This report has been produced on a short timescale and only addresses a subset of the Data Standards, with a view to eliciting further community feedback to continue the full breadth of a security review. Our findings and suggestions are presented on a best-effort basis, with the intention of contributing to a continuing process of assessment and improvement. We welcome any feedback, comments, answers, corrections, or other ongoing discussion. We would also like to thank members of the Data Standards Body for very valuable feedback on earlier drafts.

- Section 2 examines the user-facing consent process.
- Section 3 extends this analysis to authentication and security requirements in the Data Standards and compares them with best-practice and related standards including FAPI[3] [FAPI-1.0-BL, FAPI-1.0-ADV], OpenID Connect [OIDC], and the TDIF[4].
- Section 4 considers existing examples of the authentication flow in accredited CDR participants and how closely they align with the Data Standards.
- Section 5 summarises authentication flow considerations.
- Section 6 covers the Registry standards.

[1] https://github.com/ConsumerDataStandardsAustralia

[2] Partially-open projects including the Trusted Digital Identity Framework and the COVIDSafe app have a reasonable amount of openly-available material but constrained, confidential or non-existent public feedback, making it very difficult to understand and examine decisions.

[3] https://openid.net/wg/fapi/

[4] https://www.digitalidentity.gov.au/tdif

- Section 7 compares the Registry standards with comparable standards for client authentication.
- Section 8 compares the current standards, as they apply to consent, with the relevant legislation and identifies where updates need to be made.
- Section 9 provides a summary of Recommendations and Suggestions.

In some cases, we have made specific recommendations. For other issues, when the best option was less clear or there were significant other trade-offs related to usability, finance, or other issues, we have flagged a suggestion for further discussion.

1.2 Summary of findings

The main conclusion of Sections 2 and 3 is that a single one-time password (OTP) is not (any longer) a best-practice method of authenticating consumers. This is particularly acute for payment functions, but also relevant to consent to share highly sensitive personal data. When we consider the detailed knowledge of the authentication protocol, the level of vigilance required from the consumer, and the possibility that the OTP may be sent via a channel that a plausible attacker can eavesdrop, the risk of data leakage is high. Fortunately, there are usable alternatives that do not introduce unwarranted friction and do not significantly increase the risk of phishing. For example, in the banking industry a very large number of consumers have now installed a bank's app on their phone, which could very easily be used to authenticate the customer at the Data Holder in a way that would be much more secure than a web-based password and login or an OTP. This method is already in effective use in the UK's open banking scheme.

We recognise, however, that this recommendation is controversial, and may not be feasible for other CDR-connected industries such as the energy sector. We have therefore also made a series of suggestions for attempting to reduce the risk of the OTP process.

Our analysis of the Registry functions and protocols finds them to be generally well designed—we have made a small number of technical suggestions for improvement or consideration.

Section 8 points out one important drafting issue in the legislation.

1.3 Sections of the standards that we did not have time to examine

This report does not include analysis of:

- the consent process for secondary account holders (e.g. joint accounts),
- consent revocation, and the ADR-hosted Revocation Endpoint,
- DR and software accreditation.

A complete list of Data Standards sections we covered is in Appendix A.

1.4 Next steps

We hope this report contributes to a continued positive direction for the Data Standards—not as a certification of perfection (which is never possible for something as complex as this), but as an honest assessment and a set of suggestions for specific improvements, that contribute to a continuing process of refinement and improvement.

Studies of this kind should be a regular, ongoing part of any standards process, because attacks and assumptions change constantly.

2 Security analysis of consumer consent

This section aims to define the attacker model for CDR user authentication and data-sharing authorisation.

The defined authentication flow is as follows:

1. The first stage is a Consent Request phase, in which the Accredited Data Recipient (ADR) asks for the user's consent to request data from the Data Holder.

2. In the second phase, the user is redirected to the Data Holder and asked to enter a one-time password (OTP) sent via a pre-registered communication channel such as SMS or email.

3. In the third phase, the Data Holder describes to the user what data will be shared and asks for final confirmation.

Some security properties, including phishing prevention, depend on consumers not being tricked into diverging from the proper authentication flow, so we shall consider carefully exactly what "knowing the flow" means and what assumptions we can make about the user's knowledge and behaviour. User behaviour communication becomes part of the attack model. Assume that the attacker can trick the user into doing anything that you did not effectively communicate to the user they were not supposed to do.

Note that FAPI 2.0 has an attacker model [2], which is explicit about the standard sorts of internet-based attackers, but much less clear about what the user can be tricked into doing, except that they can be redirected to arbitrary websites.

For the user-facing consent mechanisms, however, the Data Standards' threat model is not obvious or standard.

We assume the attacker is expected to know or be able to guess:

- the person's customer identifier at the Data Holder; and

- the person's email address, phone number or other method of OTP delivery (for sending, not reading).

Clearly in scope

The attacker may control:

1. an unauthorised entity pretending to be an ADR; or
2. a compromised ADR, attempting to extract more data than the user gave consent for; or
3. a user, who may have some access to a target user's phone or account, attempting to trick an ADR into divulging the other user's data.

Compromised Data Holders and a compromised Registry are outside the threat model. However, attacks in which malicious parties attempt to spoof the Registry or Data Holder, or in which they misdirect or hijack the Authentication and Authorisation steps, are in scope.

Clearly out of scope

The authentication flow is obviously not intended to defend against an attacker who controls:

1. the user's login credentials for the Data Holder; or
2. a compromised ADR and also the person's OTP delivery method; or
3. the person's OTP delivery method alone, if an honest ADR simply relays the information.[5]

In each of these cases there is an obvious attack. For example, in Case 3, if the attacker controls the target person's OTP delivery method, they request data transfer via the correct protocol at the honest ADR, and enter the targeted user's customer ID number (or other requested customer identifier) when directed to the Data Holder. Since they can read the OTP, they can submit it to the Data Holder, then the Data Holder will transfer the data to the honest ADR. If the ADR immediately shares a copy of the targeted consumer's data, then the attacker can read it.

Similarly, Case 2 is clearly out of scope, though note that it's not out of the question: large organisations such as Telstra and Google often control many people's communications (SMS and email), and may also be potential ADRs. Similarly, there are common scenarios in which one person (an employer or partner) has access to another person's email or phone.

[5] There is an accredited site that does exactly this: https://mycdrdata.regionalaustraliabank.com.au/

2.1 Attacks

All the attacks described in this section are detectable by a diligent user who knows exactly what flows are permitted. However, in some cases they require a great deal of vigilance and background knowledge to prevent. Section 3 considers attacks on the OTP that are undetectable by the consumer.

We understand that the CDR Rules require each Data Holder to provide a dashboard that shows what has been shared [CDR, Consent Standards], so these attacks are all detectable after the fact if the user visits the Data Holder's dashboard. However, it is not likely to be obvious whether the ADR was malicious, or whether the consent step was intercepted by another malicious party. Hence it is not clear whether the ADR should be de-accredited.

2.1.1 Attacks by a compromised ADR

The attack here would be to extract extra data, assuming that the user intended to consent to some data sharing.

A compromised ADR could misdirect the user to an Authentication and Authorisation sequence that they control. That is, one that looks exactly like that of the Data Holder, but is delivered from a different website under the ADR's control.

This would obviously be detectable by a diligent user (by checking the URL), but could be made very convincing-looking to casual users, even those familiar with their Data Holder's website.

The compromised ADR could pass the user's account number and OTP through to a real session with the Data Holder, in which the data being requested was much more than the user thought they were consenting to.

Another possibility is to simply ask the user to enter their OTP into the ADR's website or app. This would obviously be detectable by a user who understood the flow, but may not be suspicious to a user who did not know what to expect. Encouraging ADR's to specify what the flow will be may provide some degree of training to consumers who are using the CDR regime regularly. However, it will not address first time users who have not previously interacted with a genuine ADR. Such users will be susceptible to misinformation from a malicious ADR, or site claiming to be an ADR. For non-first time users, whether such subtle training will persist is also open to debate. Studies have indicated that users do not actively read all of the content on a web page[6], as such, the subtle guidance as to how the authentication flow is supposed to work may not even be read or remembered by the consumer.

A variation on this would be an ADR app that uses an embedded user-agent inside the app—this makes it prohibitively difficult for the user to see what URL they visit. For this reason, FAPI prohibits the use of embedded user-agents [FAPI-1.0-BL, §7.5] by requiring the best practice defined in BCP 212 is followed, which dictates that when a native app is requesting an authorisation it

[6] https://www.nngroup.com/articles/how-users-read-on-the-web/

must call out to a web browser, rather than using an embedded browser within the app:

> Section 9 of OAuth 2.0 [RFC6749] documents two approaches for native apps to interact with the authorization endpoint. This best current practice requires that native apps MUST NOT use embedded user-agents to perform authorization requests and allows that authorization endpoints MAY take steps to detect and block authorization requests in embedded user-agents. ([BCP-212, §8.12])

Yet another variation is that the ADR app could direct the embedded browser to the real Data Holder site, let the user enter the OTP, and then either synthesise the 'OK' click on the consent screen, or hide all of the Data Holder's consent screen except the 'OK' button.

Whether it is possible for an Data Holder to detect that their authorisation endpoint is being accessed through an embedded user-agent remains an open question. If the non-compliant party using the embedded user-agent has performed it naively, i.e., without specifically trying to hide the fact, it should be easy to detect. However, were a malicious party to actively hide the fact an embedded user-agent is being used, it would seem unlikely that an Data Holder could detect that without using sophisticated browser fingerprinting. Therefore there is further reliance on the consumer knowing that they should not authenticate to the Data Holder from within the ADR app, and knowing (and being able to distinguish) the difference between an embedded user-agent and a standard web browser app.

The information currently provided to consumers does not explicitly warn against authentication to the Data Holder from within the ADR app. In one consumer story on the CDR website[7], specifically the "Save money: get help with your personal budgeting" example, it sounds like it is permitted:

> "Using the app, Penny gives Pennypinchers consent to share data from her bank account, which is separate to the joint account she shares with Carl. Penny is redirected from the app to her internet banking page. Penny then receives a One Time Password from her bank, which she types into the app to authorise her bank to share her transaction data with Pennypinchers."

In particular, the reference to entering the OTP into the app would appear to be incorrect, as this should be entered into the browser. This is extremely unhelpful given the crucial importance of consumers understanding that they should *never* enter their OTP into the ADR app. If instead this example is intended to demonstrate the use of the bank's app as the point of entry, that should be made much clearer.

More broadly, providing consistent educational material across ADRs, DHs, and the CDR regime itself will be important to effective user understanding. If

[7] https://www.cdr.gov.au/resources/consumer-data-right-stories

understanding of the authentication flow is critical to the security of it, then producing standardised guides, potentially in the form of short videos, which can be included on DHs and ADRs websites could help deliver a clear and consistent message for users.

2.1.2 Attacks by a malicious party pretending to be an ADR

Obviously the system is not intended to defend against a corrupt ADR who controls the user's method of OTP delivery (see above). However, there are several less severe versions of this model.

In this attack, a malicious party would pretend to be an ADR (without necessarily trying to spoof any particular ADR). We assume that the consumer (perhaps mistakenly) thinks they want to share information with this entity, perhaps because they do not realise that it is unaccredited, or perhaps because they intend to consent to only limited data sharing. The attacker performs an attack very similar to that described in Section 2.1.1, tricking the user into divulging their OTP. In the background, the attacker would initiate an interaction with an honest ADR as if they were the target user attempting to transfer data from an honest DH. When the attacker tricks the consumer into divulging their OTP, the attacker replays it at the (honest) DH, with the aim of extracting information about the user via the honest ADR.

This works if the ADR displays information to the user in the belief that it is the user's data. For example, the Regional Australia Bank's website[8] simply produces a PDF for the user (in the belief that it is that user's data).

2.2 Conclusion

The main conclusion of this section is that, although an OTP-based authentication flow seems simple for users, it actually requires great vigilance and considerable background knowledge. Users who do not know *exactly* what OTP flows are permitted may very easily have their consent bypassed.

Recommendation 1. *Consumers should be clearly warned that they need to check the URLs of their Data Holder OTP entry, even if they have been directed there by a trusted source. They should also be informed that the OTP entry should never be via an ADR's website or app.*

Suggestion for further discussion 2. *Consider ways to raise awareness of the existing list of current providers*[9]*. The existing CX requirement for ADRs to provide a link is good*[10]*, but needs to be supported by clear messages so that consumers know to be suspicious of a purported ADR that doesn't provide the link.*

[8] https://mycdrdata.regionalaustraliabank.com.au

[9] https://www.cdr.gov.au/find-a-provider

[10] https://d61cds.notion.site/Collection-and-use-consents-fcf5e47455274d26b028d218b22f017a #33

We will discuss suggestions for uplifting authentication methods later, but even if such improvements are made, there will always be some requirement to communicate to consumers how to avoid being tricked into divulging their authentication credentials. Some of these attacks would still work if, for example, consumers were required to enter their passwords and their second factor of authentication—a malicious ADR could attempt to control them too. So something like the above recommendations for user communication will still be required even if the authentication procedure is strengthened.

3 Comparison of Data Standard Requirements in Competition and Consumer (Consumer Data Right) Rules 2020 with related standards and best-practice: consumer authentication

This section examines the CDR standards in comparison with best practice and related standards including OAuth 2.0 [OAUTH, 4, 5, 6], OpenID Connect [3, 7, 9, 10, OIDC], FAPI 2.0 [FAPI-1.0-BL, FAPI-1.0-ADV] and NIST's Digital Identity Guidelines [NIST-SP800-63B].

Competition and Consumer (Consumer Data Right) Rules 2020, Division 8.4 requires that the Data Standards Chair make one or more standards for aspects related to the CDR. In particular, 8.11(1)(c)(i), which requires "authentication of CDR consumers to a standard which meets, in the opinion of the Chair, best practice security requirements:". It is therefore necessary to evaluate whether the proposed authentication method meets "best practice security requirements".

The authentication standard prohibits the use of existing credentials, in particular existing passwords. Instead it mandates a single factor one-time password of between 4 and 6 digits is transmitted over an existing channel that the consumer has previously set up with the Data Holder. Some possible channels include SMS and email.

In terms of international best practice the prescribed authentication methodology would not meet minimum standards. The EU's Payment Services Directorate (PSD2) mandates Strong Customer Authentication, involving at least two authenticator factors [PSD2, Articles 4(30) and 97]. It is a common misconception that PSD2 applies only to payments. Whilst payment initiation has gained much media coverage, PSD2 also defines data access through Account Information Services (AIS) [PSD2, Annex I] and their related Account Information Consents. As such, the AIS workflow is much more akin to banking data under the CDR. Crucially, the Strong Customer Authentication requirements apply equally to the Payment Initialisation Service and the Account Information Service [PSD2, Articles 97(4)].

Looking more broadly than open banking, the prescribed authentication methodology would not meet NIST standards for authenticators, as defined in *NIST Special Publication 800-63B Digital Identity Guidelines: Authentication*

and Lifecycle Management. We shall look more closely at this standard in regards to the references for Authenticator Assurance Levels [NIST-SP800-63B, §4] in subsection 3.2.

Closer to home, the ACSC's Information Security Manual [ISM] provides Australian best practice recommendations. Whilst it should be noted that, as a standard rather than an implementer, the Data Standards are not required to meet the ISM recommendations; however, the providers that will have to implement the Data Standards are recommended to satisfy the ISM recommendations, and may internally be required, through policy or committent, to meet ISM recommendations. As such, evaluating whether the Data Standards are in conflict with ISM recommendations is worthwhile for prospective implementers, as well as being useful for evaluating whether the Data Standards meet best practice. The ISM recommends multi-factor authentication, with passphrases as a last resort if that is not possible:

1. Multi-factor authentication is used to authenticate unprivileged users of systems. [ISM, ISM-0974]
2. Multi-factor authentication is used to authenticate users accessing important data repositories. [ISM, ISM-1505]
3. When systems cannot support multi-factor authentication, single-factor authentication using passphrases is implemented instead. [ISM, ISM-0417]

This means that a bank or other provider implementing the Data Standards may not be able to meet ISM recommendations.

3.1 Restrictions on Authentication

Restricting authentication methods used in the OIDC Hybrid Flow to only a single factor OTP presents a number of problems. Firstly, this is not best practice security; second, it does not provide strong authentication of the customer; and third, it may repurpose a communication channel for a purpose other than the one for which it was set up.

3.1.1 Best Practice

As discussed above, best practice requires multi-factor authentication. The proposed OTP would more conventionally be considered a second factor, and potentially a weak second factor if delivered via email or SMS. The ability to intercept email or SMS is greater and it is for this reason that even when used in a multi-factor setting the ISM recommends "...authentication factors that involve something a user has should be used as part of multi-factor authentication" [ISM, Multi-factor authentication]. In order for the OTP channel to be considered something a user has it must be delivered via a medium that is tied to specific device. As such, email would not be considered to meet this requirement. For channels using the public switched telephone network (PSTN)

such as SMS [NIST-SP800-63B, §5.1.3.1], NIST states that "[i]f out-of-band verification is to be made using the PSTN, the verifier SHALL verify that the pre-registered telephone number being used is associated with a specific physical device" [NIST-SP800-63B, §5.1.3.3]. How such verification takes place is not specified, but it would appear to be challenging.

3.1.2 Strong Authentication of the Customer

Strong Customer Authentication is defined within the EU's PSD2 regulations, requiring multi-factor authentication [PSD2, Article 4(30)]. Whilst the specifics of the PSD2 are out of scope, the notion of strongly determining the authentication of the customer is essential if the action being taken is that of providing consent to access what constitutes private and potentially sensitive data about the customer. As such, the method of authentication must ensure, within reasonable bounds, that it really is the customer, or someone the customer has consented to act on their behalf, providing the consent. Since the only authentication factor permitted under the defined OIDC Hybrid Flow is a single OTP delivered over a potentially insecure channel, such a level of authentication cannot be achieved.

For example, SMS messages are susceptible to SS7 attacks[11] and email is not considered a secure channel [NIST-SP800-63B, §5.1.3.1]. Even if external threats have been reduced, for example via network operators actively protecting against such attacks, the local or partner attack vector remains. Many people will have their phones configured to display email and SMS notifications on their lock screens[12]. As a result, mere access to a locked phone could allow a partner or member of the household access to sufficient information to impersonate that individual and consent to a CDR transfer. For example, they could register the partner with a ADR and receive the OTP to the device and access it without needing to unlock the phone.

To address such concerns, NIST expressly state that "[i]f a secret is sent by the verifier to the out-of-band device, the device SHOULD NOT display the authentication secret while it is locked by the owner" [NIST-SP800-63B, §5.1.3.1]. It is generally not possible for the sender to know or alter the configuration of the device, and therefore they cannot ascertain whether the recommendation is met. SMS permissions are also often sought, and frequently granted, to untrustworthy apps.

Whilst any potential attacker may not be able to get direct access to the data themselves, they would be able to access the summary data provided by ADRs, which may be very detailed and, even if not, could still provide insights and the potential for control by the malicious partner or household member. The same issue applies when couples separate. Can one partner continue receiving summary data if they set up the service? What reminders are sent to the

[11] https://www.mobileeurope.co.uk/why-is-ss7-still-a-security-threat/

[12] https://developer.android.com/guide/topics/ui/notifiers/notifications#lockscreenNotification

subject? Are they across multiple communication channels? Is the need to check the Data Holder's dashboard being effectively communicated to all consumers?

3.1.3 Repurposing Communication Channel

The Data Standards state that "The delivery mechanism for the OTP is at the discretion of the Data Holder but MUST align to existing and preferred channels for the customer and MUST NOT introduce unwarranted friction into the authentication process" [CDR, Authentication Flows]. This indicates that an existing channel that has already been set up between the Data Holder and the customer will be used for delivery of the OTP. However, it does not require that the customer was advised at the time of setting up the channel that it would be used for such a purpose. As this is a single factor, the security implications are very different from that of its use for general communication or as a second factor in a multi-factor authentication. It is possible multiple customers within the same household or family could share the same communication channel. For example, an email address, tablet, or phone. If the customer was not informed that the communication channel could be used as the sole authentication mechanism for accessing their data then the channel should not be repurposed without prior consent from the user.

Recommendation 3. *Require Data Holders to ask consumers for permission to use a certain channel as the CDR OTP delivery channel if it was not originally set up as an authentication channel.*

3.2 Authentication Flow

The proposed single factor OTP authentication flow presents a number of problems, both in terms of security and in terms of compliance with existing standards. As already mentioned, a single factor OTP may not meet the ISM requirements. Furthermore, it will not meet the NIST authentication standards [NIST-SP800-63B] in general, since email OTP is not recognised as a valid form of out-of-band OTP, and SMS is considered restricted—requiring alternative options to be provided.

Whether it meets the TDIF standard is ambiguous. Currently the Data Standards reference the *Authentication Credential Requirements* specification of the TDIF. The most recent version of this we could find was August 2018, version 1.3, which we believe to be the version intended to be referenced. However, this document appears to have been deprecated by the TDIF; it is not available on the TDIF website and the content from it appears to have been mostly moved into *05 Role Requirements*, §4.

05 Role Requirements has also seen significant changes; most notably it no longer defers directly to [NIST-SP800-63B] for specification of authenticator properties, unlike the *Authentication Credential Requirements* which did. Instead the TDIF has extracted parts of [NIST-SP800-63B] directly into the TDIF specification, but not completely. As such, details on restrictions and

constraints on SMS channels no longer appear to be part of the TDIF specification. Furthermore, *05 Role Requirements* no longer ties Credential Levels directly to NIST's Authenticator Assurance Levels as the referenced TDIF document did:

> NIST Authenticator Assurance Levels (AAL) equate to TDIF CLs. (*Authentication Credential Requirements*, §2.5)

As a result the current data standard does not appear to meet the authenticator requirements specified by [NIST-SP800-63B] and by extension does not meet the requirements of version 1.3 of the *Authentication Credential Requirements*, which defers directly to NIST:

> For guidance on the specific requirements for each CL applicable to the different types of credentials refer to NIST SP 800-63B. (*Authentication Credential Requirements*, §2.5)

However, it could be argued it does meet the new TDIF specification [TDIF-05] on account of the weakening in security requirements that has taken place.

Recommendation 4. *As defined in the referenced TDIF requirements [TDIF-ACR-1.3], Credential Levels are directly equivalent to NIST's Authenticator Assurance Levels [NIST-SP800-63B, §4]. Update references to use [NIST-SP800-63B] rather than the TDIF for both defining Credential Levels and authenticator properties. If the Credential Levels from the TDIF are retained, refer directly to [NIST-SP800-63B] for authenticator standards to maintain the intended security level.*

3.2.1 Usage of Legacy SMS Channels

[NIST-SP800-63B] already considers SMS out-of-band channels to be restricted due to the inherent weaknesses and history of attacks. Further expanding the usage of SMS as an authentication channels runs the risk of creating a greater ability for Smishing and other SMS scams, which have continued to be prevalent in recent years[13].

3.2.2 Enumeration Attacks

Any password or PIN is potentially subject to an attacker trying to guess it. In the case of numerical passwords, these brute-force attacks are often called "enumeration attacks" because they can be attempted by systematically testing all possible sequences of digits of the required length. (They can also, of course, be performed by random guessing.) Standard defences are twofold: first, try to inhibit the opportunity for one attacker to make a large number of guesses, second, choose the secret randomly from a large set of possibilities, so that enumerating the whole set requires a very large number of guesses. A secret

[13] https://www.anz.com.au/security/fraud-detection/latest-security-alerts/

chosen with an "entropy of 20 bits" means roughly that it is as hard to guess by brute-force as a secret consisting of 20 perfectly random coin tosses.

The current Data Standards state "The provided OTP MUST be numeric digits and be between 4 and 6 digits in length" [CDR, Authentication Flows]. However, both [NIST-SP800-63B, §5.1.3.2] and the TDIF (both old and new) explicitly state that the minimum entropy is 20 bits:

> TDIF Req: CSP-04-02-03j; Updated: Jun-21; Applicability: C[14] The Applicant MUST generate random Authentication secrets with at least 20 bits of entropy. (*05 Role Requirements*, §4.2.3)

To achieve 20 bits of entropy the provider would need to provide a numeric OTP with a minimum length of 6 digits. A 4 digit numeric OTP would only provide 14 bits of entropy. It is clear that the Data Standards currently do not meet this requirement, as the maximum number of digits the Data Standards allow is the minimum sufficient number of digits to meet this requirement.

Furthermore, it is not clear why the security standards should specify a maximum—we assume that this is for usability, rather than security, reasons. The decision to constrain the length to a maximum of 6 digits is unduly low and prevents a provider, or potentially even a consumer, from determining the risk profile for their only authentication factor—some users may be comfortable with 8-digit PINs and may prefer to set a higher level of security.

The Data Standards also state that Data Holders should attempt some defence against enumeration attacks. What sort of defences should be assumed or specified? For example, how many failed attempts should be tolerated? Should this raise an alarm per account, per request, per day, or per ADR? Furthermore, NIST requires that rate limiting must be implemented when using a generated OTP with entropy of less than 64 bits [NIST-SP800-63B, §5.1.3.2].

One kind of attack here is for a compromised ADR to attempt data extraction from multiple accounts, possibly thousands at the same Data Holder, and for each attempt make the maximum number of OTP guesses. This is unlikely to succeed for any particular account number, but may be likely to succeed at least once out of a few thousand targets. For example, if the OTP length is 4 digits and 5 guesses are permitted, then an attack on 2,000 accounts is likely to make one correct guess.

Recommendation 5. *Set a minimum OTP length of at least 6 digits and require rate limiting measures to be implemented.*

Suggestion for further discussion 6. *Consider removing the maximum OTP length and allowing Data Holders or even consumers to choose to make them longer than 6 digits.*

[14] Applicability type "C" in TDIF requirements refers to the Credential Service Provider, which binds Credentials (secrets used for authentication, such as a password or multi-factor authentication (MFA) device) to a digital identity. In the Data Standards, this role belongs to the Data Holder.

Suggestion for further discussion 7. *Consider more detailed guidance about defending against enumeration attacks, for example that Data Holders should be alert for attacks against multiple different accounts at once.*

One common method of defending against enumeration attacks is a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). These often show the user a collection of photographs and ask them to identify all the trucks, bridges, bicycles, etc. The idea is to allow the server to distinguish humans from robots, on the assumption that robots are more likely to be attempting enumeration attacks. A CAPTCHA at the Data Holder may help to mitigate enumeration attacks. However, we are cautious to recommend them, for a few reasons:

- CAPTCHAs can be replayed by sufficiently motivated attackers, for example re-displaying them on a site that the attacker controls and asking visitors to solve them in real time, then relaying the answers to the targeted Data Holder;
- CAPTCHAs introduce another privacy risk into the authentication flow, because the organisation running the CAPTCHA is now present in the browser session;
- CAPTCHAs introduce significant friction into the login experience. There are known usability, and more importantly, accessibility problems with existing CAPTCHAs[15]. They are also considered a significant waste of humanity's time[16].

Overall, we believe that uplifting authentication standards (using the recommendations in this report) would be a better tradeoff than CAPTCHAs for improving security without inconveniencing consumers.

3.2.3 Sources of Randomness for OTPs

The Data Standards do not provide any requirements or guidance for how to generate appropriately random OTPs, and in fact even the requirement that an OTP is random is a "SHOULD" requirement rather than a hard "MUST" requirement:[17]

> The algorithm for the creation of the OTP is at the discretion of the Data Holder but SHOULD incorporate a level of pseudorandomness appropriate for the use case ([CDR, Authentication Flows])

[15] https://www.w3.org/TR/turingtest/

[16] https://blog.cloudflare.com/introducing-cryptographic-attestation-of-personhood/

[17] "SHOULD" requirements need a specific justification if they are not implemented. In this case, 6 digits of entropy is not a difficult requirement to meet, and we do not see a genuine situation in which there would be a reasonable justification for not implementing it. Hence "MUST" makes more sense, to avoid confusion or argument about whether any excuse for less entropy is acceptable. This matters because there is little benefit to extending the PIN length if it is derived from predictable values such as dates of birth.

As the TDIF requirements the Data Standards defer to themselves defer to NIST, the Data Standards do not meet the NIST requirements for randomness generation:

> The verifier SHALL generate random authentication secrets with at least 20 bits of entropy using an approved random bit generator [NIST-SP800-90A]. ([NIST-SP800-63B, §5.1.3.2])

The newer TDIF requirements [TDIF-05] do not include a requirement for the use of approved sources of randomness.

An inadequate source of randomness could allow an attacker to infer information about OTP generation and potentially predict the valid OTP for an authentication flow—or at least limit the search space. This allows them to avoid the defences against enumeration attacks by reducing the number of guesses they would have to make to determine the valid OTP, increasing their chances of success.

Recommendation 8. *Align the Data Standards with NIST [NIST-SP800-90A, NIST-SP800-63B] to provide requirements for appropriate sources of randomness. Change the "SHOULD" requirement about levels of pseudorandomness to a "MUST" requirement, or defer to NIST.*

3.2.4 Capping Security

Security standards should not cap the level of security that can be implemented. A security standard should set the minimum and possibly recommend additional security, but be compatible with implementers exceeding the minimum security standards. There is good reason for this: once a security standard is fixed, it will only get weaker. As such, it should facilitate gradual evolution to stronger security measures to counter stronger attacks that naturally develop over time. Furthermore, not every person or organisation has the same risk profile. As such, consumers and organisations should be free to set the security requirements above the minimum to mitigate against perceived increased risk. For example, people in financial difficulties, people from (other) stigmatised groups, people at risk of family violence, celebrities or high net worth individuals may wish to have much stronger protection over their sensitive data.

Currently there is no consumer autonomy over the security of their CDR data. If they are concerned about the OTP-only authentication flow specified in the Data Standards, they have no mitigating strategies open to them. Likewise, organisations that support such individuals cannot implement stronger security to satisfy their customer needs. In the absence of a stronger baseline authentication flow, consumers should be provided with the power to block all CDR requests to their account to mitigate the increased risk associated with the weaker single OTP authentication flow currently proposed.

Suggestion for further discussion 9. *Require Data Holders to provide a CDR Lock that is initially on by default and prevents all CDR requests from*

being approved. Consumers can switch this lock off via their current stronger authentication method if they wish to start using CDR. Should a stronger authentication flow be permitted by default the CDR lock could remain, but default to being off.

Suggestion for further discussion 10. *Permit stronger authentication flows to be implemented and allow weaker ones to be disabled by default for user accounts that already have stronger authentication methods established.*

3.3 Phishing Protection

Forbidding the use of existing passwords for authentication appears to be a response to the potential phishing attack that could be triggered via fake ADRs, or even malicious ADRs. However, the proposal to forbid the use of existing credentials does not mitigate the risk unless one of the following assumptions holds true:

- The phishing site follows the Data Standards as well and does not ask for the username and password; or
- The customer is sufficiently knowledgable about the Data Standards to know that username and password should not be asked for by the data holder.

The latter assumption is addressed through the requirements of the consumer experience, which state that "Data Holders and Data Recipients MUST state in consumer-facing interactions and communications that services utilising the CDR do not need access to consumer passwords for the purposes of sharing data" [CDR, Authentication Standards]. However, this appears to assume that the consumer is experienced in the CDR process already. More specifically, a consumer who has never used the process before[18] and happens to land on a malicious website that falsely claims to be an accredited CDR data recipient will not have seen such instructions before. The malicious operator can then state the exact opposite and redirect to a phishing site to socially engineer the consumer into providing their full credentials.

As such, neither of these assumptions can be assured to be met. The first can be dismissed entirely, since the phishing site has no reason to follow the restrictions of the Data Standards. The second seems highly unlikely since it will require an in-depth understanding of authentication methods and restrictions with the standard, and at the very least experience in CDR and a sufficiently high level of attentiveness. (See Section 4 for some discussion of the communication difficulty.)

In the absence of either of the above assumptions, the single factor OTP workflow may still create an increased phishing risk for the primary credentials, whilst also compromising on the quality of the authentication for consent to

[18] Or a consumer who has not attentively read all text during previous CDR interactions, which seems likely to be a common case.

access CDR data. Users will become accustomed to being redirected to the data holder for authentication and there is no reason to assume they will appreciate the subtlety of the use of only a OTP.

Alternative approaches for mitigating the risk of phishing could include WebAuthn[19], although we recognise this could be a significant shift in capability for participants.

Recommendation 11. *Ensure messaging about constraints is consistent across providers and publicise those constraints outside of the CDR authorisation flow so that users are educated before starting the process about what to expect and reject.*

3.4 Levels of Assurance

The Data Standards determine the an LoA of 2 is required for read operations, translating to a Credential Level of CL1 [CDR, Levels of Assurance (LoAs)]. The TDIF describes the following Credential Levels:

> **Credential Level 1 (CL 1):** provides a low level of confidence that the Individual controls a Credential bound to their Digital Identity. The intended use of this level is for services where the risks of getting Credential binding wrong will have negligible to minor consequences to the Individual or the service.
>
> **Credential Level 2 (CL 2):** provides a medium level of confidence that the Individual controls a Credential bound to their Digital Identity. The intended use of this level is for services where the risks of getting Credential binding wrong will have moderate to high consequences to the Individual or the service.
>
> **Credential Level 3 (CL 3):** provides a very high level of confidence that the Individual controls a Credential bound to their Digital Identity. The intended use of this level is for services where the risks of getting Credential binding wrong will have very high consequences to the Individual or the service.
>
> (*05A Role Guidance*, Credential Levels)

This raises the question of whether banking data, in particular a year's worth of transaction data, should be classified as CL1, CL2, or CL3, and by extension whether the current classification of CL1 is appropriate.

3.4.1 Credential Level of CDR Data

Banking data, and in particular transaction data, should be considered to be some of the most identifying [11], valuable, and potentially sensitive data held about a person. Different people have different preferences for data privacy, but

[19] https://www.w3.org/TR/webauthn-3/

a person's banking transactions may be as important to their privacy as location, communication or health data. Because people tend to engage in financial transactions as part of almost every activity, transaction data can provide a snapshot across a person's entire life. The nature of some transactions allows a recipient to infer potentially sensitive attributes, which are afforded stronger protection under the Privacy Act. In particular, subscriptions or payments for membership dues could reveal things such as:

- membership of a political association
- membership of a professional or trade association
- membership of a trade union

Furthermore, transactions involving online dating sites, or special interest subscription websites, could reveal sexual preferences or practices. Medical information may also be able to be inferred through medical charges, prescription charges, or payments to cosmetic surgeons.

Quite apart from explicitly protected sensitive attributes, transaction data also provides information about location, through the metadata associated with electronic payments, as evidenced by the suggestion to use card payments during the pandemic to locate people[20]. This is by no means the first such instance of using card transactions to track and monitor people's movements[21].

Because of its extensive connections to other kinds of data and activities, transaction data may also be extremely useful for re-identifying other data about the person, by linking payments from the transaction dataset to matching activities in some other data.

As such, the importance of strong consent for the transfer of banking data is essential, as the consequences for unintended transfer could be high for the individual. Similar arguments can be made for other industries which handle sensitive data and will be covered by the Data Standards, such as insurance. However, the Data Standards make no allowance for different levels of sensitivity across the different industries it covers and instead applies a blanket requirement for the Credential Level across all relevant industries. As such, it is difficult to see how a classification of CL1 can be justified for CDR transaction data. At the very least a classification of CL2 should be adopted, and there may be arguments for CL3 for the more sensitive industries such as banking; however, that may be beyond what is possible to deploy as some banks may not currently meet that level even for their full logins. At minimum, the authentication requirements and Credential Level for CDR data access for an industry should not be lesser than that required for general account access within that industry.

If CDR data were to be classified as CL2, the current proposed Authentication Flow would fail to meet the TDIF requirements, in particular the lack

[20] https://www.smh.com.au/politics/federal/card-payment-data-to-be-used-to-track-people-in-coronavirus-hotspots-20201113-p56eea.html

[21] It has been anecdotally reported that one of Australia's big four banks used EFTPOS data analysis to evaluate customer movements during one of the Melbourne's White Night festivals.

of multi-factor authentication and the prohibition on using a memorised secret. The classification of the data should be justifiable independently from what is consistent with any proposed Authentication Flow.

Recommendation 12. *The default Credential Level in the Data Standards should be a minimum of CL2. Allowance can be left for industry-wide exceptions in the case that there is a strong argument that an industry does not handle sensitive data, but it is unclear if such an exemption would ever apply.*

While there are industries which lack in digital maturity and therefore may struggle to immediately meet such a requirement, these industries should be encouraged to uplift their security rather than lowering the security of the Data Standards to make allowance. However, encouraging such industry-wide general security uplift is clearly outside the scope of the Data Standards, and therefore we offer no formal suggestion or recommendation for this.

4 Overview of existing UX in Australia and the UK

The CDR standards contain several points that constrain authentication flow options for reasons that are not directly related to the security of the authentication flow, but represent tradeoffs with other criteria such as usability and protection against phishing.

We are not usability experts, but we include here a short examination of the existing authentication flows for banking, including examples from Australian banks' CDR implementations, and generic examples from UK open banking. We did not actually try any of these flows—we simply looked at online descriptions of them.

We find a complicated set of relationships between usability, security and standards-compliance, which justifies a much more explicit and separate analysis of which usability standards are required, and why. It does not seem helpful for the standards to preclude more secure alternatives, particularly when some parts of some higher-assurance processes are already being offered.

This section also reinforces the difficulty of communicating some aspects to consumers: we find that two Australian banks do (sometimes) require consumers to log in with their username and password, though not at the Data Holder site that the ADR directs the consumer to. This greatly increases the subtlety of the message that needs to be communicated to consumers: they may be asked for a password, just not at the website they are explicitly redirected to. We were confused on this point, and we think that the current standards are unclear. If phishing defence relies on this distinction being successfully communicated to ordinary consumers, that communication is going to be a significant challenge.

4.1 CDR requirements that do not directly relate to authentication security

The Data Standards (Consumer Experience) state:

> **Authentication:** Passwords
>
> Data Holders and Data Recipients MUST state in consumer-facing interactions and communications that services utilising the CDR do not need access to consumer passwords for the purposes of sharing data. ([CDR, Authentication Standards])

The Data Standards (Security Profile) also include (emphasis ours):

> - Data Holders MUST NOT request that the customer enter an existing password *in the redirected page*
>
> ([CDR, Authentication Flows])

These two requirements are not the same—the first precludes any need for password-based authentication, but the second prohibits it only on the page that the ADR has redirected the consumer to. An instruction to go and log in to the Data Holder's online banking portal, or a Data Holder app on the person's phone, seems inconsistent with the first requirement but not the second.

Clearly the difference needs to be resolved and clearly communicated to consumers. One suggestion is to put all the things that consumers need to know in the Consumer Experience section, while reserving the Security Profile only for rules that improve security regardless of consumer understanding.

There are a number of other requirements in the Authentication Flows section that limit the options for more secure authentication:

> - The delivery mechanism for the OTP is at the discretion of the Data Holder but MUST align to existing and preferred channels for the customer and MUST NOT introduce unwarranted friction into the authentication process
> - ...
> - The provided OTP MUST be used only for authentication for CDR based sharing and MUST NOT be usable for the authorisation of other transactions or actions
>
> In line with CDR Rule 4.24 on restrictions when asking CDR consumers to authorise disclosure of CDR data, unwarranted friction for OTP delivery is considered to include:
>
> - the addition of any requirements beyond normal data holder practices for verification code delivery
> - providing or requesting additional information beyond normal data holder practices for verification code delivery
>
> ([CDR, Authentication Flows])

The distinction between the OTP and a verification code is important: while the OTP is specific to the CDR ecosystem, a verification code is the one-time password typically used as a means of multi-factor authentication when logging into a bank normally.

4.2 What Australian banks are currently doing

This section briefly surveys the CDR flows we see already in use by Australian banks. Our intention is not to judge whether these are compliant with the standard, but to ask whether the standards are working to guarantee secure authentication flows. Existing practice may not always be secure, but it generally represents a good basis for assessing what is practically usable.

Many banks use a simple OTP delivery method: NAB sends SMS[22]; Bank Australia sends email[23]; Bank of Queensland allows either SMS or email[24]; IMB uses SMS[25]; and Bendigo Bank uses SMS[26]. We were unable to tell from a brief Internet search what Regional Australia Bank, Westpac, or ANZ use.[27]

The Commonwealth Bank requires login to their app or online banking system[28]. Their instructions are shown in Figure 1. Consumers are required to log in to Internet banking or a phone app in order to read their OTP, which (we think) is then entered into the website the ADR directs the consumer to. On the one hand, this is more secure than sending the OTP via SMS or email, because of the stronger authentication method required to access it; on the other hand, it is less secure than simply giving consent directly through the app or online banking portal, because the OTP-entry step still happens outside the secure authentication and is still at risk of enumeration attacks.

We do not think this could constitute "normal data holder practices for verification code delivery" because, by definition, customers do not need to be logged in already in order to get the verification code to complete their normal login—if they did need to be logged in to get the verification code, they would be unable to log in at all.

In summary, it would be better if the consumer, having authenticated strongly with the bank via their app or online banking portal, could then give their consent for CDR data sharing there. This would be both more convenient and more secure than what the Commonwealth Bank is currently doing, and it does not make sense that the bank's choices or the CDR standards should preclude it.

The Bank of Melbourne asks for a normal login using standard Internet Banking credentials—see Figure 2

[22] https://www.nab.com.au/personal/customer-support/open-banking

[23] https://bankaust.com.au/support/open-banking#sts=Our%20Consumer%20Data%20Right%20policy

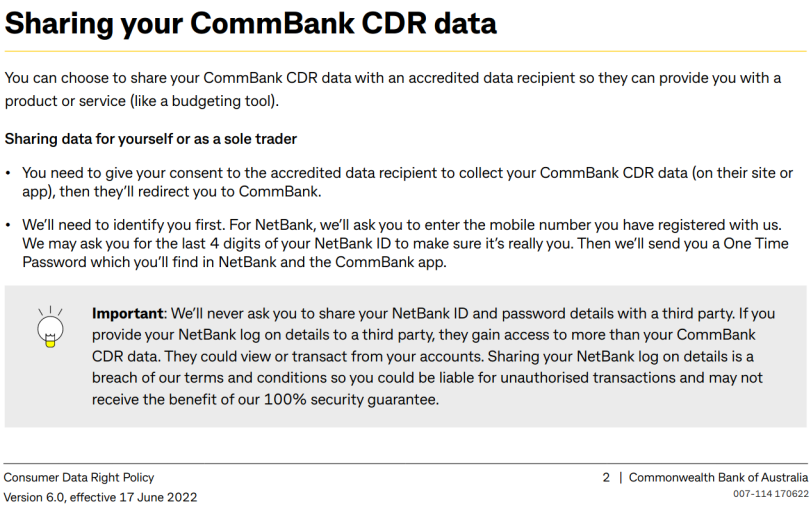
[24] https://www.boq.com.au/personal/banking/openbanking

[25] https://www.imb.com.au/openbanking#setup

[26] https://www.bendigobank.com.au/open-banking/#HowToShare

[27] We believe Westpac and ANZ use SMS.

[28] https://www.commbank.com.au/content/dam/commbank/security-privacy/consumer-data-right-policy.pdf



Sharing your CommBank CDR data

You can choose to share your CommBank CDR data with an accredited data recipient so they can provide you with a product or service (like a budgeting tool).

Sharing data for yourself or as a sole trader

- You need to give your consent to the accredited data recipient to collect your CommBank CDR data (on their site or app), then they'll redirect you to CommBank.
- We'll need to identify you first. For NetBank, we'll ask you to enter the mobile number you have registered with us. We may ask you for the last 4 digits of your NetBank ID to make sure it's really you. Then we'll send you a One Time Password which you'll find in NetBank and the CommBank app.

Important: We'll never ask you to share your NetBank ID and password details with a third party. If you provide your NetBank log on details to a third party, they gain access to more than your CommBank CDR data. They could view or transact from your accounts. Sharing your NetBank log on details is a breach of our terms and conditions so you could be liable for unauthorised transactions and may not receive the benefit of our 100% security guarantee.

Figure 1: Commonwealth Bank CDR authentication instructions. Source: https://www.commbank.com.au/content/dam/commbank/security-privacy/consumer-data-right-policy.pdf. Last accessed 24 June 2022.

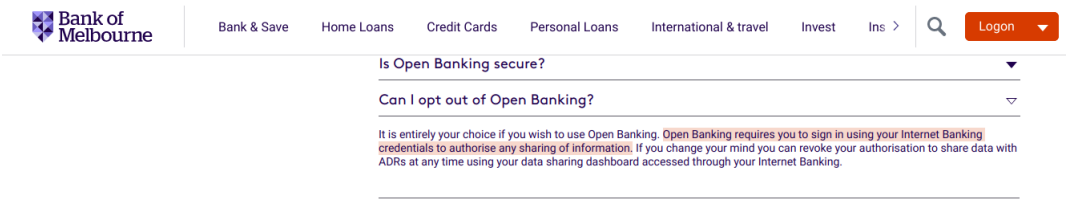


Figure 2: Bank of Melbourne CDR authentication instructions. Source: https://www.bankofmelbourne.com.au/online-services/open-banking. Last accessed 24 June 2022.

HSBC seems to use their existing Secure Key, which may be either physical or digital.[29] Based on the online description, this does not seem to be distinct from the code that is usable for the authorisation of other transactions.

4.3 Open banking Authentication flows in the UK

The UK Open Banking flows generally include a passcode and second factor of authentication—Figures 3 and 4 show browser-based and app-based flows respectively. Consumers are required to enter their passcode, which is their long-term password, using randomly selected digits to decrease the risk of phishing.

[29] https://www.hsbc.com.au/help/open-banking/faq/

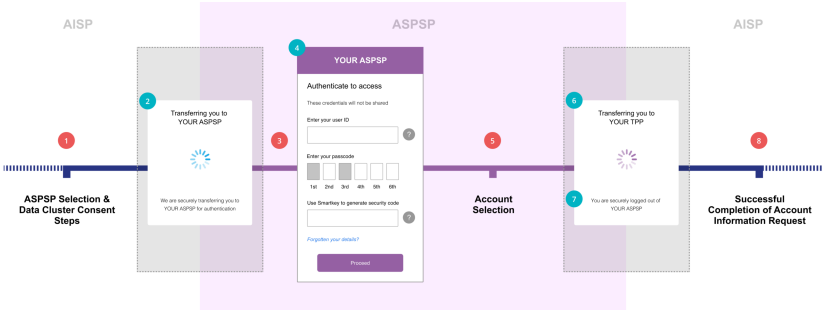


Figure 3: UK Open banking, browser-based authentication flow. Note the two factors of authentication. Source: https://standards.openbanking.org.uk/customer-experience-guidelines/authentication-methods/redirection/browser-based-redirection-ais/latest/

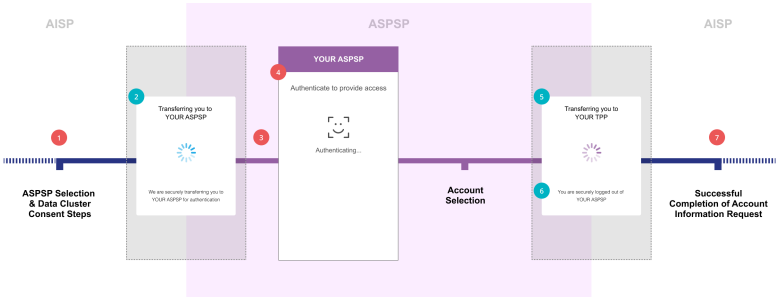


Figure 4: UK Open banking, app-based authentication flow. The consumer uses whatever authentication they normally use to log in to the app. Source: https://standards.openbanking.org.uk/customer-experience-guidelines/authentication-methods/redirection/app-based-redirection-ais/latest/

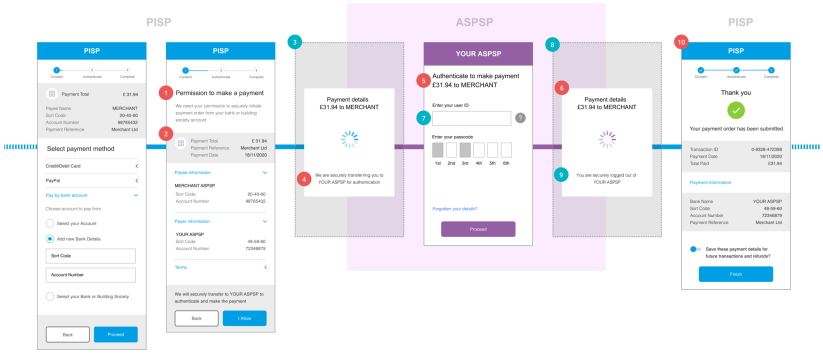


Figure 5: UK Open banking flow for the case of special exemption from two-factor authentication. Note that the consumer still needs to enter (some digits of) a long-term password. Source: https://standards.openbanking.org.uk/customer-experience-guidelines/authentication-methods/rts-sca-exemptions/aspsp-applies-an-available-exemption/latest/

Only when there are specific exemptions is a single factor allowed—this flow is shown in Figure 5. Even then, the secret is the person's long-term password, not an OTP.

We have not conducted a systematic examination of increased UK phishing risk as a consequence of these decisions, but we do not see any reason to believe that Australian banking would be any more susceptible than UK banking. It does not seem that requiring password-based authentication presents an unacceptable risk for phishing.

5 Summary: Consumer authentication flow

Best practice guidelines require multi-factor authentication. If phishing remains a concern there are better ways of addressing it than reducing the quality of the authentication. For example, an app based authentication/consent flow could be implemented that involves no web-based redirection, and therefore does not create a workflow where the customer is redirected to the data holder for authentication. Many things have changed since the decision to adopt a single OTP was made in 2018—many consumers now have the option of logging in on a (bank or other) app on their phone, and many are familiar with the 2-factor authentication flow of myGovID[30], Digital iD[31], and similar systems. This would be much more secure, and may be easier for consumers than keeping track of all the checks they need to do to ensure that the OTP flow is being correctly followed.

Australian banks already seem to be implementing some app-based or portal-

[30] https://www.mygovid.gov.au/
[31] https://www.digitalid.com/

based logins—if this is not meant to be compliant with the standard, then communicating that to consumers is going to be challenging. If it is sufficiently usable to be compliant with the standard, then allowing authenticated consumers to give CDR consent directly when logged in seems better than an OTP, for both security and usability.

Consider how other jurisdictions have approached this problem. The UK requires Strong Customer Authentication, using at least two factors from different types of authentication. It also requires a consumer to be able to authenticate themselves using the same method they would normally use.[32] This allows the consumer to set their own risk profile and therefore have autonomy over their security posture. As such, a high-risk consumer can choose to employ biometrically protected cryptographic hardware, WebAuthn[33], or a combination of both to secure access to their bank and any associated open banking data. Conversely, the CDR caps the strength of the permitted authentication by forcing use of a single factor out-of-band OTP that is delivered through an "existing channel". As such, a consumer who has setup multi-factor authentication to access their account is forced to use a weaker single factor authentication method for accessing CDR data.

There appears to be no way for a high-risk or conscientious consumer to either raise the security posture of access to their data, or opt-out of the possibility entirely by blocking all such requests. The OTP authentication method may be much more easily intercepted than their usual authentication mechanism, denying them autonomy over their security and putting them at greater risk.

Recommendation 13. *Consider alternative authentication flows that provide a higher level of consumer authentication without exacerbating phishing risk, for example, an app-based two-factor authentication flow.*

6 Security analysis of the Registry

This section examines the security of the Register APIs, specifically those for Client Authentication and Registration. Overall, this part of the standard seems well designed. We did not find any serious problems, though we did have some suggestions for relatively minor improvements (such as key length in one case) and some questions about details not specified in the standard.

6.1 Public Key Infrastructure

The standard defines a 3 tier PKI:

1. Root Certificate Authority - 4096 bit RSA, Expires: 17/10/39

[32] https://standards.openbanking.org.uk/customer-experience-guidelines/authentication-methods/latest/

[33] https://w3c.github.io/webauthn/

2. Intermediate Certificate Authority - 2048 bit RSA, Expires: 17/10/34, Max-Path-Length: 0

3. To be issued Data Recipient and Data Holder Certificates

The root CA certificate is at the upper-end of acceptable lifetime (20 years), although not outside of what is regularly seen in industry. The risks associated with such a long life are mitigated by the larger key size of 4096 bits and by the expectation that the Root CA private key should be held offline due to its relatively rare usage.[34]

The intermediate CA has a 15 years lifetime, but only a 2048 bit key size. Such a long lifetime for a key that will be in regular use and with what would be considered a moderate key size is not desirable. The key's regular use on an internet-connected server means that there is at least some risk of key compromise, and the long lifetime means that there is some risk that keys of that length may be successfully factorised even by conventional computers within its lifetime. Best practice recommendations specify a key size in excess of 2048 bits to be assured of security into the 2030's[35]. The ACCC Certificate Policy[36] states the key will be either 2048 bits or 4096 bits. Since both must be supported, selecting the larger key size would be more appropriate. A larger key size would not resolve all concerns related to the expected activity associated with the private key in issuing certificates as the CDR scheme ramps up. A shorter lifetime would provide stronger assurance.

Recommendation 14. *Specify a longer key length or shorter lifetime for the intermediate CA key, in keeping with best practice recommendations.*[37]

The intermediate CA is correctly configured to limit path length. Path length determines the number of subordinate certificate authorities that are permitted below the intermediate CA in the hierarchy. In this instance the intermediate CA has a path length of 0, which indicates it is an issuing CA and cannot have a further subordinate CA below it. Verification of compliance with the path length restriction is the responsibility of the party performing certificate verification.

Information about the expected lifetime of leaf certificates, i.e., those issued to CDR participants, appears in the ACCC Certificate Policy[38] and seems to specify 3 years for subscriber certificates. The expected primary purpose of these certificates will be for establishing MTLS connections. Current best practice for

[34] The appropriate lifetime of a certificate is based on a number of factors including the resistance of the underlying key to brute-force attack and the manner and frequency in which the underlying key is used. The more a key is used or if it is used during an automated process, the greater the risk it could be compromised.

[35] https://www.keylength.com/en/compare/

[36] https://www.cdr.gov.au/sites/default/files/2020-12/CDR - ACCC Certificate policy.pdf

[37] https://www.keylength.com/en/compare/

[38] https://www.cdr.gov.au/sites/default/files/2020-12/CDR\ -\ ACCC\ Certificate\ policy.pdf

TLS certificates is for them to have a lifetime of 13 months or 398 days[39] [40] [41]. Browser restrictions do not directly impact on the CDR regime as the MTLS connections will be established outside of a browser context. However, it should be anticipated that increasingly TLS libraries will start to validate based on the 398 day limit. Furthermore, the same concerns associated with the lifetime of TLS certificates apply to MTLS connections used in the CDR regime. Whilst real-time checking of revocation status helps to reduce the time-lag between compromise detection and blocking, it does not help reduce the window for possible exploitation in the case of an undetected compromise.

As such, it would be advisable to follow industry best practice and limit the leaf certificates issued by the CDR CA to 398 days.

Suggestion for further discussion 15. *Consider limiting the lifetime of leaf certificates to 398 days.*

6.1.1 Key Rotation

A policy for key rotation is advisable. The exact approach for determining an appropriate period is context specific, involving evaluations on storage, usage, and key length. NIST publishes standards providing guidance on how to determine an appropriate period [NIST-SP800-57-pt1-r5].

Some caution is required when undertaking key rotation if the corresponding certificate is, or could be, hard-coded or pinned into applications. This will be the case for the root certificate authority and in this context could be true for the intermediate certificate authority as well, since there is only a single intermediate CA. In such circumstances any key rotation would require any apps that hard-coded the certificate to be updated prior to the rotated key being used. This is less of an issue in a server-to-server setting since such updates can be coordinated and rolled out, which should be the case in this instance. There should be no reason to need to perform MTLS from client apps, which would present a greater challenge in achieving roll out and uptake of updates.

Suggestion for further discussion 16. *Review [NIST-SP800-57-pt1-r5] for determining crypto periods and key rotation policy. Publish that policy and include notification periods that will be used when performing key rotation. Establish notification process to warn participants of when a key rotation will take place.*

6.1.2 Consequence of limiting certificate lifetime

Irrespective of the exact limit imposed, the issue of certificate expiration should be considered in the standard. FAPI 1.0 requires the use of sender constrained

[39] https://chromium.googlesource.com/chromium/src/+/ae4d6809912f8171b23f6aa43c6a4e8e627de784

[40] https://blog.mozilla.org/security/2020/07/09/reducing-tls-certificate-lifespans-to-398-days/

[41] https://support.apple.com/en-us/HT211025

access tokens [FAPI-1.0-ADV, §5.2.2(5) and §5.2.3(1)], which should also be extended to refresh tokens. Where that sender constraint is achieved through embedding a hash of the MTLS certificate, the lifetime of the access token will effectively be bounded by the lifetime of the MTLS certificate. With refresh token lifetime defined in the standard as being equal to the consent period, which could be up to a year, sooner or later the lifetime of a refresh token will exceed the lifetime of the certificate. This is discussed in more detail in Section 7.

6.1.3 Certificate Validation

The certificate validation provided in the standard is minimal, whereas best practice would be to explicitly state all fields/aspects of the certificate that need to be validated. There are multiple degrees of separation between a developer reading the Data Standards and a statement of what needs to be validated in the certificate.

The Data Standards provide a list of validation components [CDR, Certificate Management]:

1. Checking for certificate validity
2. Issuer-to-subject name chaining
3. Policy and key use constraints
4. Revocation Status

Point 1 is a catch-all that duplicates point 4, i.e., it is not revoked. However, it does not mention essential validation checks like checking the valid from and valid to dates.

Point 3 states: "Each certificate has the applicable and appropriate x.509 certificate extensions, e.g. CA and CRL signing, Digital Signing, Client and Server Authentication, etc". However, it is left to the developer to know what the applicable and appropriate values for those extensions should be. It would be better to explicitly state that the CA certificates must have CA extension set and the leaf certificates must not.

The Data Standards provide a link to the ACCC Certificate Validation document for further information[42]. However, this reference also does not explicitly state all the fields that need checking to perform certificate validation. It states "...confirm the validity of each certificate in the certificate path in accordance with IETF PKIX standards". However, it does not provide a link or reference to which specific standard should be used. This creates some ambiguity and leaves it up to the developer to find the appropriate standard.

The rest of the validation points are the same as in the Data Standards. As such, a developer will have to read through two documents, merge them,

[42] https://cdr-support.zendesk.com/hc/en-us/articles/900005826963-Certificate-Validation

and then find a third and merge that to get a complete list of what should be validated in a certificate. Given that the path of least resistance for a developer implementing certificate validation is to do no validation, it would be advisable to make accessing a clear specification of what to validate as easy as possible.

Recommendation 17. *Provide (or link to) a single, complete, set of instructions for certificate validation.*

There is also a duplication of information in the ACCC Certificate Validation[43] document in the *Unavailability* section in which the first two sentences are effectively duplicated:

> While certificate status services are designed to be available 24x7 without interruption, there may be times when the certificate status services are unavailable.
>
> Relying Parties are bound to their obligations and the stipulations of the Relaying Parties, Certificate Policy and the Certification Practice Statement irrespective of the availability of the certificate status service.
>
> Certificate status services are available 24x7 without interruption. However, there may be times when the certificate status services are unavailable.
>
> Relying Parties are bound to their obligations and the stipulations of the Relying Party Agreement, Certificate Policy and the Certification Practice Statement irrespective of the availability of the certificate status service.[44]

Recommendation 18. *Revise Certificate Validation document to remove duplication opting for the first two paragraphs. The duplicates (latter two) contain a contradiction in themselves. If the certificate status services* ***are*** *available 24x7 without interruption there cannot by definition be times when they are unavailable.*

6.2 Mutual TLS

The Data Standards require the use of Mutual TLS (MTLS) for back-channel communication directly between the ADR and DH:

> All back-channel communication between Data Recipient Software Product and Data Holder systems MUST incorporate, unless stated otherwise, [MTLS] as part of the TLS handshake:

[43] https://cdr-support.zendesk.com/hc/en-us/articles/900005826963-Certificate-Validation

[44] https://cdr-support.zendesk.com/hc/en-us/articles/900005826963-Certificate-Validation

- The presented Client transport certificate MUST be issued by the CDR Certificate Authority (CA). The Server MUST NOT trust Client transport certificates issued by other authorities.
- The presented Server transport certificate MUST be issued by the CDR Certificate Authority (CA). The Client MUST NOT trust Server transport certificates issued by other authorities.

> End points for transferring CDR Data that are classified as not requiring authentication do not require the use of [MTLS]. ([CDR, Transaction Security])

While the exact endpoints which require MTLS are noted in the Security Endpoints section, the relationship between these two sections is left implicit. The Certificate Management section further muddies the water here by stating "ADRs may choose to secure their endpoints with a Register CA issued certificate or a certificate issued by a public CA", which includes multiple endpoints used in back-channel communication with the DH—while it can be determined from reviewing all endpoints listed in the Security Endpoints section that no ADR-hosted endpoints use MTLS, this conclusion requires careful reading.

Recommendation 19. *The above quoted section should clarify the following:*

- *The exact list of endpoints requiring MTLS is presented in the Security Endpoints section.*
- *MTLS is only required for DH-hosted endpoints and Register-hosted endpoints which require authentication.*

This could be additionally clarified through the inclusion of tables summarising the MTLS, Client Authentication, and Bearer token requirements for the endpoints hosted by each participant.

Recommendation 20. *The word "endpoints" in the above quote from the Certificate Management section is a link to a non-existent fragment `#end-points`. This should likely lead to the Security Endpoints section, which has the correct fragment `#security-endpoints`.*

6.3 Registry endpoints

The Data Standards indicate that the Register's JWKS endpoint is at `/jwks` [CDR, Register APIs]. Using the base URLs from the same section, this would be `https://api.cdr.gov.au/jwks` for the TLS endpoint. Accessing this URL results in a 404 Not Found response—the JSON Web Key Set for the Register is currently located at `https://api.cdr.gov.au/cdr-register/v1/jwks` instead, which differs from the documented endpoint. Similarly, the OpenId Provider Config endpoint indicates that it is at `/.well-known/openid-configuration`; however, the expected URL, `https://api.cdr.gov.au/.well-known/openi`

d-configuration, does not contain the config document and instead gives a 404 Not Found response.

It is unclear whether the documented endpoints or the current production endpoints are correct, but at least one of them is wrong.

Recommendation 21. *The documented JWKS and OpenId Provider Config endpoints and the equivalent endpoints in the production Register API should be aligned, such that the documented endpoints are valid in the context of the production API.*

6.4 Validation of JWT signatures using Software Statement Assertions

When an ADR registers with a DH, the JWT that is provided to the Dynamic Client Registration endpoint includes a Software Statement Assertion (SSA) from the CDR Register and is signed by the ADR [CDR, Client Registration]. The SSA is a nested JWT signed by the Register and contains the JWKS URI of the ADR, which specifies the location of the public key used to verify the signature on the JWT signed by the ADR. The exact order of the steps that should be taken to verify the JWT are not specified precisely, but the inferred order is as follows:

1. Decode the ADR JWT into JSON.
2. Extract the SSA JWT from the `software_statement` field.
3. Request the Register's JWKS from the Register's JWKS endpoint.
4. Verify the signature on the SSA JWT using the appropriate JWK from the Register's JWKS.
5. Decode the SSA JWT into JSON.
6. Extract the ADR's JWKS endpoint URI from the `jwks_uri` field.
7. Request the ADR's JWKS from the ADR's JWKS endpoint.
8. Verify the signature on the non-decoded ADR JWT.

These verification steps are inside-out, requiring that the content of the ADR JWT is decoded and used to reach a point where verification of the already-used ADR JWT's signature is possible. Cryptography best practice dictates that signature verification should always be the first step carried out,[45] but this is clearly not possible here—and, in fact, verifying the outermost signature even requires verifying a second, nested signature first.

We do not think this causes a vulnerability, because an attempt to fake this registration will be detected when the last signature fails to verify. However,

[45] See for example XML signature verification guidelines: https://www.w3.org/TR/xmldsig-bestpractices/

there are two reasons that it is best practice to do the signature verification first: first, that it excludes non-authenticated participants from even submitting their data, which matters if there is a buffer overflow or similar security problem in any of the other data processing, and second, that it reduces the time spent by the server on a fake attempt.

Suggestion for further discussion 22. *This process is derived from the Open Banking UK registration profile*[46]*, which itself extends the* OAuth 2.0 Dynamic Client Registration Protocol*. As the RFC specifies that Registration requests must use the* `application/json` *content type [OAUTH-DCR, §3.1] while the approach taken by OBUK and the Data Standards requires the use of a* `application/jwt` *content type, it is not clear that this is a valid extension of the RFC.*

Consider an approach which is a valid extension of the RFC while also providing a signature on both the SSA and the data from the ADR. One such approach may be to require that the request object is presented as a raw JSON object which contains two fields: the `software_statement` *field defined in the RFC [OAUTH-DCR, §3.1.1] and already used by the Data Standards, and another field containing the signed JWT from the ADR with the addition of a new sub-field* `ssa_hash`*, which contains a hash of the SSA it accompanies*[47]*.*

This approach would allow the signatures to be verified in order and the processing to be immediately abandoned on a failed verification, while also preserving the binding between the SSA and ADR's JWT through the use of the `ssa_hash`*.*

6.5 Transitions between different authorisation states

The Data Standards show the opportunity to transition between states [CDR, Participant Statuses]. For example, Data Recipients may transition among Active, Suspended, Revoked, or Surrendered states. However, there is not currently any detail about how these transitions happen, which is particularly important for the transition from Revoked or Suspended states back to Active—it is important that an attacker cannot compromise an ADR, get their status Revoked, and then somehow persuade the Registry to return them to Active status.

Potential attacks could include:

- compromising the transition process, e.g. by compromising the accounts of the people who implement it,
- compromising how the status values propagate out to the CDN (DNS hijack or such).

[46] `https://bitbucket.org/openid/obuk/src/master/uk-openbanking-registration-profile.md`

[47] The approach used for hashing the SSA could be derived from the very similar `at_hash` defined in OpenID Connect for binding an ID Token to the accompanying Access Token [OIDC, §3.1.3.6].

Suggestion for further discussion 23. *In consultation with ACCC, specify procedures (whether electronic or human-mediated) for authorisation state transitions. The transition from Revoked or Suspended back to Active is particularly challenging, because the decision to Revoke may have been motivated by credential compromise.*

7 Comparison of Data Standard Requirements in Competition and Consumer (Consumer Data Right) Rules 2020 with related standards and best-practice: ADR, Data Holder and Registry authentication

7.1 Sender Constrained Tokens

The FAPI Advanced Security Profile, and by extension the Data Standards, is ambiguous about whether both access tokens and refresh tokens should be sender constrained. The requirement for sender constraint on access tokens is made explicit in 5.2.2. of the FAPI Advanced Security Profile [FAPI-1.0-ADV, §5.2.2]. However, the same requirement for refresh tokens is not explicit, although it once was. The wording was changed[48] partly in response to an issue originating from the CDR standard[49] and discussed in more detail on the FAPI issue tracker[50]. The issue correctly identified that there was a problem in requiring the authorization code to be sender or holder-of-key constrained when that was a browser-based endpoint and would require the MTLS client certificate to be installed in the browser. However, in addressing that issue the reference to refresh tokens appears to also have been removed, but without justification or explanation.

Having a sender-constrained access token but not a sender-constrained refresh token makes little sense in this context. The access token has a maximum life of 10 minutes and as such, the window for potential compromise and exploitation is short. However, the refresh token is long-lived, and since the prohibition on rotation, potentially even longer lived. The risk of refresh token theft is therefore greater, which is exactly what sender constraining is supposed to address.

This view is reflected in the draft OAuth 2.0 security recommendations, which makes explicit the requirement to sender constrain both access and refresh tokens [8, §4.13.2].

Recommendation 24. *Recommend that FAPI restores the requirement for refresh tokens to be sender constrained. In the meantime, specify explicitly that*

[48] https://bitbucket.org/openid/fapi/pull-requests/178
[49] https://github.com/ConsumerDataStandardsAustralia/infosec/issues/31
[50] https://bitbucket.org/openid/fapi/issues/202/authorization-code-and-refresh-token-must

refresh tokens should be sender constrained.

7.1.1 Certification Lifetime and Sender Constrained Refresh Tokens

If refresh tokens are sender constrained this raises a further issue with regards to lifetime of the tokens and the underlying certificate used to verify the sender constraint. Irrespective of what lifetime the MTLS certificates issued by the CDR CA are, there will come a point at which a consent period, and therefore lifetime of a refresh token, extends beyond the life of the current MTLS certificate.

The obvious way to address this would be to use the Distinguished Name of the certificate to constrain the sender, as described in the use case of multiple clients sharing the same key in section 8.10 of the FAPI Advanced Security Profile [FAPI-1.0-ADV, §8.10]. If this is to be the approach taken then it should be explicitly described in the CDR standard.

If the Data Holder instead constrains based on the actual key material, rather than the certificate data, then it will not be able to handle updated MTLS certificates and will reject future requests due to either an invalid MTLS certificate or a breach of the sender constraint.

Note, adopting the approach taken in 8.10 of the FAPI Advanced Security Profile [FAPI-1.0-ADV, §8.10] places an additional burden on the CDR CA to ensure that the Distinguished Name is unique per accredited participant[51] and that it has been fully verified, and therefore places a greater trust burden on the CDR CA. It will also require that the Distinguished Name, or part thereof, remains static between certificate renewals. Participants may wish to see greater certificate transparency to protect them from any potential impersonation attacks associated with a miss-issuance of a certificate.

Recommendation 25. *Define how certificate expiration will be handled by the MTLS sender-constrained tokens. Update specifications as necessary in terms of what should be being checked during verification and how it is to be used to enforce the sender constraint.*

7.2 Token Rotation

The issue of token rotation has been a problem for other open banking jurisdictions. However, there appear to be flaws in the discussion on the FAPI issue tracker[52]. There appears to be a belief that presence of client authentication, through for example, MTLS, negates the need for refresh token rotation as there is no replay attack. However, the replay attack remains in the case of the client being compromised, since both the refresh token and the private key used for

[51] The ACCC CDR Certificate Policy uses the term participant to describe those issued certificates. In this instance we are referring to anyone participating in the CDR regime, who is accredited, and has been issued a certificate signed by the ACCC CA.

[52] https://bitbucket.org/openid/fapi/issues/456/proposal-should-we-remove-support-for

client authentication could have been compromised. This would allow a malicious third-party to obtain an access token silently and without detection by the legitimate client. Refresh token rotation would detect that as the legitimate client would be locked out, or the compromised token would become invalid. One scenario provides detection, one prevention, but both are desirable. This dramatically reduces the potential size of the window between compromise and exploitation. Without any such rotation, and refresh tokens with a life of up to 12 months, that window is extremely large.

The discussion on the issue tracker even outlines the correct resolution to the problem, which is to allow an overlap between old and new refresh token until the new refresh token is used. This extends the window for attack, but still bounds it, particularly if regular access is occurring, as would be the case in many CDR scenarios.

The underlying issue is that the OpenID Connect protocol fails to correctly synchronise the rotated token between client and server. If the response containing the new token was lost in transmission the client would be out of sync and unable to authenticate using the token. That wasn't a big issue in user authentication because it just falls back to existing credentials (username and password). However, for automated authentication in a delegated machine-to-machine setting, such as the CDR, this becomes a problem because the user is no longer in the loop. Removing token rotation rather than resolving the possible loss of synchronisation is a compromise on security, and would be unnecessary if the root cause was addressed.

The CDR has taken the decision to prohibit refresh token rotation from September 16th 2022 (FAPI 1.0 Migration Phase 2). This in part appears to be based on discussions about what would be recommended in FAPI 2.0, described in more detail in Decision 209[53]. However, FAPI 2.0 has not outright prohibited the rotation of refresh tokens; it has only prohibited rotation in the absence of appropriate synchronisation error mitigation. Specifically it currently states Authorization servers "shall not use refresh token rotation unless, in the case a response with a new refresh token is not received and stored by the client, retrying the request (with the previous refresh token) will succeed" [FAPI-2.0-SEC, §4.3.1.1]. The standard does discourage their use, but under an incorrect assumption that there is no security benefit: "This specification discourages the use of this feature as it doesn't bring any security benefits for confidential clients, and can cause significant operational issues" [FAPI-2.0-SEC, §4.3.1.1] which as discussed above is not correct. Conversely, clients are required to support refresh token rotation as specified in point 6 of Section 4.3.2.1 of the FAPI 2.0 Security Profile [FAPI-2.0-SEC, §4.3.2.1].

We recognise that the Data Standards are simply following the direction of some other jurisdictions and FAPI in general, albeit a possibly undesirable direction.

Suggestion for further discussion 26. *Recommend that FAPI re-evaluates the security implications of not performing refresh token rotation if a confidential*

[53] https://github.com/ConsumerDataStandardsAustralia/standards/issues/209

client is compromised.

Recommendation 27. *If no changes are forthcoming to FAPI, the Data Standards should be consistent with it and only discourage but not prohibit token rotation.*

7.3 Refresh Token Inconsistencies

The current text will become inconsistent when the FAPI 1.0 Migration Phase 2 comes into force. Currently it reads:

> Refresh Token expiration MAY be any length of time greater than 28 days but MUST NOT exceed the end of the duration of sharing consented to by the Consumer.
>
> Until September 16th 2022:
>
> Data Holders MAY cycle Refresh Tokens when an Access Token is issued. If Refresh Token cycling is not performed then the Refresh Token MUST NOT expire before the expiration of the sharing consented by the Customer.
>
> From September 16th 2022 (FAPI 1.0 Migration Phase 2):
>
> Data Holders MUST NOT cycle refresh tokens (rotation). In other words, Refresh Tokens SHOULD be issued with an "exp" equal to the sharing duration authorised by the Customer.

The proposed statement is neither consistent with the statement above, nor is it consistent with what came before. The minimum requirement of 28 days could be longer than the consent period, which as defined in 4.11(1)(b) of the legislation allows single occurrence consents or a period specified up to 12 months [1, §4.11(1)(b)]. The Refresh token should, or possibly, must be issued with an expiration date the same as that as the end date consented to by the customer. The issue of MTLS certificate expiration may also impact on the appropriate end date.

Recommendation 28. *If Data Holders MUST NOT cycle refresh tokens then refresh tokens MUST be issued with an "exp" equal to the sharing duration authorised by the Customer.*

7.4 Client Authentication for Protected Endpoints

While the CDR makes use of Mutual TLS as a Holder of Key mechanism, it deviates from FAPI 1.0 [FAPI-1.0-ADV, §5.2.2] and does not allow MTLS to be used for client authentication. Instead, it specifies two allowed client authentication approaches based on signed JWTs: `private_key_jwt`, as specified in OIDC [OIDC, §9] and used by FAPI 1.0 [FAPI-1.0-ADV, §5.2.2]; and its own authentication technique using self-signed JWTs.

The `private_key_jwt` client authentication is consistent with the OIDC specification, with the exception of a minor change to the recommended value

of the `aud` claim: OIDC recommends that the value should be that of the authorisation server's Token Endpoint, while the CDR recommends the use of the authorisation server's issuer identifier URL, with the issuer identifier, Token Endpoint, or endpoint being accessed as valid alternatives. These adequately identify the intended audience, so this change is unlikely to cause issues. The CDR does not specify a maximum expiry time for this JWT.

The self-signed JWT client authentication does not appear to be a standard approach, presumably because there is normally no requirement for an OIDC Client (ADR) to authenticate the OIDC Authentication Server (DH). The self-signed JWT client authentication is provided as a JWT Bearer token in the Authorization request header, and the JWT content is effectively the same as the JWT presented in `private_key_jwt` client authentication. While `private_key_jwt` client authentication specifies the location of the public keys that should be used to verify the JWT signature, self-signed JWT client authentication does not specify where the keys should be found. This makes signature validation for self-signed JWTs as written in the specification ambiguous.

Recommendation 29. *The CDR standard should explicitly specify the location of the JWKS used to verify the JWT signature in self-signed JWT client authentication.*

8 Changes in the legislation and their implications for the standards as they apply to consent

In this section we revisit the CDR legislation, particularly recent changes to the rules around consent, and consider which parts of the CDR standards may need to be updated. We are not lawyers, and the legal language is much less precise than the technical standards we have been discussing in previous sections, so please regard this section as a work-in-progress with some speculative suggestions, rather than a definitive analysis.

The consent framework for the CDR has changed considerably since the CDR was first proposed. It now provides a more granular consent model, with the addition of explicit consent required for de-identification of CDR data.

The CDR Data Standard provides a minimal definition for how consent is to be recorded and codified, primarily through the use of scopes. As such, it appears that the DH only receives the consent requests related to its specific functionality, i.e., it receives a request for collection consents and a time limit on that consent. All other types of consent are recorded by the ADR. There is a requirement for the accredited person to provide a "CDR Receipt" to the consumer when a consent is received, amended, or withdrawn. However the security properties of that receipt are not defined and its delivery is left up to the accredited person, with the only restriction being that it must be in writing and via a channel other than the consumer dashboard.

As such, there is no cryptographically bound record of the consents that the consumer agreed to, except the collection consents sent to the DH. There appears to be no requirement that the receipt is digitally signed or that it must be delivered in a manner in which the sender cannot determine whether or not it has been read.

By way of an example, it appears it would be legitimate for an ADR to provide a download URL in a push notification for the consumer to retrieve their receipt. This would allow the ADR to know whether the consumer has downloaded a copy of the receipt and therefore the scope for modifying the consents, and possibly even the receipt, without the user's permission, knowledge, or ability to prove otherwise.

The splitting of the consents into disjoint sets introduces further challenges. In particular, how are sets of consents on the DH and the ADR linked? If consents are given to the ADR, but the collection consent at the DH is refused, how are the consents that were already given to the ADR cancelled?

For example, if a consumer makes a series of wide ranging use consents to the ADR, is redirected to the DH, but then is uncomfortable with the scope of the collection consent and denies it, how is the deletion of the previously given use consents handled? This may not seem important in the simple case of no past or future relationship because a use consent without data is by definition useless. However, if a consumer has an existing arrangement or creates a future arrangement, could the orphaned consents become active? For example, suppose a consumer already has consented to wide ranging collection but for a narrow use. The consumer subsequently starts a second CDR request with the same ADR but for different uses—this time wide ranging—but changes their mind at the point of DH authorisation. An honest ADR would naturally delete the orphaned consents and do nothing further. However, a malicious ADR could assert that the consents have been given and could be applied to the data collections that were previously consented to. Without a cryptographically-bound link between the two sets of consents the consumer will not be able to prove they did not provide those consents, since they did, albeit not through to completion. Does the consumer have to explicitly revoke those consents?

More broadly, the use of a Data Receipt that is not cryptographically bound provides little ongoing assurance to the consumer. Their ability to enforce it is unclear. If the full consent set had been sent to the DH there would be a signature over the request and therefore a commitment by the DH to the consent set. It could then be held by the DH to provide a trusted store of past consents that the consumer could use if they ever had to enforce them. Furthermore, it would have prevented any risk of splitting the consent sets and negated the need for the CDR Receipt. (We discussed a little among the authors whether the DH was the right trustee for this information, given the likely competitive relationship between the DH and the ADR—it is not clear that this is the best solution, but it is worth considering the best way that a consumer can be sure of retaining a trustworthy record.)

Suggestion for further discussion 30. *Consider more detailed specifications*

for how a consumer can be assured of a binding and detailed receipt for the consents that they have granted, possibly one that uses CDR Arrangement IDs and links a specific collection consent to other ADR consents.

8.1 Issues with language in Competition and Consumer (Consumer Data Right) Rules 2020, Compilation 7, Compilation Date: 1 February 2022

This section raises some questions outside the scope of this report, but worthy of mention given the potential impact. We understand that correcting drafting errors in the legislation is not within the remit of the Data Standards Chair, but it seems impossible to make proper standards given that the legislation itself is extremely confusing. We would appreciate guidance on where to direct these observations. Our analysis is based on the current version of the rules [1].

The language in "Subdivision 7.2.3 (7.5) - Meaning of permitted use or disclosures that do not relate to direct marketing" is contradictory and ambiguous. In particular:

- "(aa) in accordance with a current **use consent** [our emphasis], de-identifying the CDR consumer's CDR data in accordance with the CDR data de-identification process and:

  (i) using the de-identified data for general research; or

  (ii) disclosing (including by selling) the de-identified data;"

- "(e) disclosing (by sale or otherwise), to any person, CDR data that has been de-identified in accordance with the CDR data de-identification process;"

In the case of (aa) we think this is a drafting error. It would make more sense if "use consent" was replaced with "de-identification consent." It does not make any sense that a use consent would allow de-identification and sale, particularly when there is a separate de-identification consent that otherwise is not used.

In the case of (e) it appears to be the same as (aa)(ii), in that disclosure of de-identified data is a permitted use, but this time without the need for even a use consent. It appears (e) would therefore supersede (aa)(ii). Such a permitted use should surely be guarded by being in accordance with a de-identification consent.

In summary, we believe that both (aa)(ii) and (e) read differently from their intended meaning, and that there should have been one clause allowing for the de-identification of data and the sharing or selling of the de-identified data, if the consumer has given a de-identification consent.

Without such guards the de-identification consent appears to be nullified since the equivalent permitted uses are granted with a use consent, or even without.

There appears to be another drafting error in "7.2 Rule relating to privacy safeguard 1—open and transparent management of CDR data". Paragraph (5)

states: "For subparagraphs (4)(e)(ii) and (g)(ii), the further information is...". There is no subparagraph (4)(e)(ii), nor (g)(ii). It appears it should be (4)(j)(ii) and (4)(l)(ii). There is also a subparagraph h(i) and h(iii) but no h(ii).

9 Conclusion and Recommendations

The CDR standards must simultaneously meet somewhat-conflicting requirements: to secure the consent process, to secure the data, to avoid exacerbating security risks to Data Holder accounts, to facilitate ease of use, etc. Although we have concentrated on the security and privacy aspects of the standards (and do not have expertise in usability or regulation), we recognise that many of the decisions in the security standards represent attempts to find a good tradeoff among different objectives.

Some parts of the security standards are actually intended to address one of these other objectives, for example setting a maximum length of an OTP is (we assume) for usability reasons, while the prohibition against logins is for the protection of Data Holder accounts rather than for the security of CDR data flows. It might be clearer if the reasons for each requirement were made explicit, particularly for those not directly related to security. For example, the OTP length could have a specified minimum (for security) and, separately, a specified maximum with a clear statement that this was for usability, rather than security reasons.

It may also be worth considering different options for different industries, given their different likely levels of account security. For example, in the banking industry, it may now be feasible to expect that a very large fraction of customers have the Data Holder's app installed on their phone. In that context, it may be appropriate to mandate app logins (with a biometric or username/password), returning the OTP to its more standard role as a second authentication factor. Consumers could then give CDR consent within the app after authentication. Conversely, in industries where app use is less common but phishing attacks are less likely to be lucrative (such as industries that do not involve payments) it may make sense to remove the prohibition against username/password authentication. Of course, expectations would then have to be carefully communicated to users.

Suggestion for further discussion 31. *Consider making non-security requirements and tradeoffs explicit, in order to allow for concrete analysis of the tradeoffs. If a requirement serves a purpose other than authentication security, make sure that its reason is explained clearly.*

9.1 Recommendations about the process

The open, community-focused standards process through which the Data Standards are developed is a leading example for other parts of the Australian government to follow. Our technical analysis was greatly aided by the complete and open history of not only the decisions that were made, but the discussions

and tradeoffs leading up to those decisions. We hope this open process is maintained, because it will lead to a successful set of standards that can continually develop in response to new threats and opportunities.

Detailed security analyses like this should be a regular part of an ongoing process of improvement, conducted ideally by a variety of people with complementary skill sets. In each case, the existing practice of publishing the reports for community feedback should be maintained, so that those directly working on implementing the standards can improve their understanding (and correct any mistakes by the consultants).

The idea of ongoing security review is mentioned in the ISM. Although the specific recommendations apply to applications, not standards, the general idea would have similar value for standards:

- **Control: ISM-1238**
  Threat modelling is used in support of application development.
- **Control: ISM-0402**
  Applications are robustly tested for security vulnerabilities by software developers, as well as independent parties, prior to their initial release and following any maintenance activities.
- **Control: ISM-1754**
  Security vulnerabilities identified in applications are resolved by software developers.

([ISM, Revision: 4, Updated: Mar-22])

In this setting the resolution would be done by the Data Standards Body, rather than software developers.

It is hard to put an exact number on the frequency or cost of these sorts of examinations, but they are worth repeating whenever substantial changes are made to the Data Standards. They should also be frequent enough that substantial changes in the external environment can be considered. This examination was very time-constrained and, as a result, we were not able to cover all aspects in depth. It may therefore make sense to initiate a longer and more in-depth review fairly soon, particularly when considering the question of whether to change recommendations about the main flow.

9.2 Summary of Recommendations

Recommendation 1. *Consumers should be clearly warned that they need to check the URLs of their Data Holder OTP entry, even if they have been directed there by a trusted source. They should also be informed that the OTP entry should never be via an ADR's website or app.*

Suggestion for further discussion 2. *Consider ways to raise awareness of the existing list of current providers*[54]*. The existing CX requirement for ADRs*

[54] https://www.cdr.gov.au/find-a-provider

to provide a link is good[55]*, but needs to be supported by clear messages so that consumers know to be suspicious of a purported ADR that doesn't provide the link.*

Recommendation 3. *Require Data Holders to ask consumers for permission to use a certain channel as the CDR OTP delivery channel if it was not originally set up as an authentication channel.*

Recommendation 4. *As defined in the referenced TDIF requirements [TDIF-ACR-1.3], Credential Levels are directly equivalent to NIST's Authenticator Assurance Levels [NIST-SP800-63B, §4]. Update references to use [NIST-SP800-63B] rather than the TDIF for both defining Credential Levels and authenticator properties. If the Credential Levels from the TDIF are retained, refer directly to [NIST-SP800-63B] for authenticator standards to maintain the intended security level.*

Recommendation 5. *Set a minimum OTP length of at least 6 digits and require rate limiting measures to be implemented.*

Suggestion for further discussion 6. *Consider removing the maximum OTP length and allowing Data Holders or even consumers to choose to make them longer than 6 digits.*

Suggestion for further discussion 7. *Consider more detailed guidance about defending against enumeration attacks, for example that Data Holders should be alert for attacks against multiple different accounts at once.*

Recommendation 8. *Align the Data Standards with NIST [NIST-SP800-90A, NIST-SP800-63B] to provide requirements for appropriate sources of randomness. Change the "SHOULD" requirement about levels of pseudorandomness to a "MUST" requirement, or defer to NIST.*

Suggestion for further discussion 9. *Require Data Holders to provide a CDR Lock that is initially on by default and prevents all CDR requests from being approved. Consumers can switch this lock off via their current stronger authentication method if they wish to take the risk and start using CDR. Should a stronger authentication flow be permitted by default the CDR lock could remain, but default to being off.*

Suggestion for further discussion 10. *Permit stronger authentication flows to be implemented and allow weaker ones to be disabled by default for user accounts that already have stronger authentication methods established.*

Recommendation 11. *Ensure messaging about constraints is consistent across providers and publicise those constraints outside of the CDR authorisation flow so that users are educated before starting the process about what to expect and reject.*

[55] https://d61cds.notion.site/Collection-and-use-consents-fcf5e47455274d26b028d218b22f017a #33

Recommendation 12. *The default Credential Level in the Data Standards should be a minimum of CL2. Allowance can be left for industry-wide exceptions in the case that there is a strong argument that an industry does not handle sensitive data, but it is unclear if such an exemption would ever apply.*

Recommendation 13. *Consider alternative authentication flows that provide a higher level of consumer authentication without exacerbating phishing risk, for example, an app-based two-factor authentication flow.*

Recommendation 14. *Specify a longer key length or shorter lifetime for the intermediate CA key, in keeping with best practice recommendations.*[56]

Suggestion for further discussion 15. *Consider limiting the lifetime of leaf certificates to 398 days.*

Suggestion for further discussion 16. *Review [NIST-SP800-57-pt1-r5] for determining crypto periods and key rotation policy. Publish that policy and include notification periods that will be used when performing key rotation. Establish notification process to warn participants of when a key rotation will take place.*

Recommendation 17. *Provide (or link to) a single, complete, set of instructions for certificate validation.*

Recommendation 18. *Revise Certificate Validation document to remove duplication opting for the first two paragraphs. The duplicates (latter two) contain a contradiction in themselves. If the certificate status services* ***are*** *available 24x7 without interruption there cannot by definition be times when they are unavailable.*

Recommendation 19. *The above quoted section should clarify the following:*

- *The exact list of endpoints requiring MTLS is presented in the Security Endpoints section.*
- *MTLS is only required for DH-hosted endpoints and Register-hosted endpoints which require authentication.*

Recommendation 20. *The word "endpoints" in the above quote from the Certificate Management section is a link to a non-existent fragment* `#end-points`*. This should likely lead to the Security Endpoints section, which has the correct fragment* `#security-endpoints`*.*

Recommendation 21. *The documented JWKS and OpenId Provider Config endpoints and the equivalent endpoints in the production Register API should be aligned, such that the documented endpoints are valid in the context of the production API.*

[56] https://www.keylength.com/en/compare/

Suggestion for further discussion 22. *This process is derived from the Open Banking UK registration profile*[57]*, which itself extends the* OAuth 2.0 Dynamic Client Registration Protocol*. As the RFC specifies that Registration requests must use the* `application/json` *content type [OAUTH-DCR, §3.1] while the approach taken by OBUK and the Data Standards requires the use of a* `application/jwt` *content type, it is not clear that this is a valid extension of the RFC.*

Consider an approach which is a valid extension of the RFC while also providing a signature on both the SSA and the data from the ADR. One such approach may be to require that the request object is presented as a raw JSON object which contains two fields: the `software_statement` *field defined in the RFC [OAUTH-DCR, §3.1.1] and already used by the Data Standards, and another field containing the signed JWT from the ADR with the addition of a new sub-field* `ssa_hash`*, which contains a hash of the SSA it accompanies*[58]*.*

This approach would allow the signatures to be verified in order and the processing to be immediately abandoned on a failed verification, while also preserving the binding between the SSA and ADR's JWT through the use of the `ssa_hash`*.*

Suggestion for further discussion 23. *In consultation with ACCC, specify procedures (whether electronic or human-mediated) for authorisation state transitions. The transition from Revoked or Suspended back to Active is particularly challenging, because the decision to Revoke may have been motivated by credential compromise.*

Recommendation 24. *Recommend that FAPI restores the requirement for refresh tokens to be sender constrained. In the meantime, specify explicitly that refresh tokens should be sender constrained.*

Recommendation 25. *Define how certificate expiration will be handled by the MTLS sender-constrained tokens. Update specifications as necessary in terms of what should be being checked during verification and how it is to be used to enforce the sender constraint.*

Suggestion for further discussion 26. *Recommend that FAPI re-evaluates the security implications of not performing refresh token rotation if a confidential client is compromised.*

Recommendation 27. *If no changes are forthcoming to FAPI, the Data Standard should be consistent with it and only discourage but not prohibit token rotation.*

Recommendation 28. *If Data Holders MUST NOT cycle refresh tokens then Refresh Token MUST be issued with an "exp" equal to the sharing duration authorised by the Customer.*

[57] `https://bitbucket.org/openid/obuk/src/master/uk-openbanking-registration-profile.md`

[58] The approach used for hashing the SSA could be derived from the very similar `at_hash` defined in OpenID Connect for binding an ID Token to the accompanying Access Token [OIDC, §3.1.3.6].

Recommendation 29. *The CDR standard should explicitly specify the location of the JWKS used to verify the JWT signature in self-signed JWT client authentication.*

Suggestion for further discussion 30. *Consider more detailed specifications for how a consumer can be assured of a binding and detailed receipt for the consents that they have granted, possibly one that uses CDR Arrangement IDs and links a specific collection consent to other ADR consents.*

Suggestion for further discussion 31. *Consider making non-security requirements and tradeoffs explicit, in order to allow for concrete analysis of the tradeoffs. If a requirement serves a purpose other than authentication security, make sure that its reason is explained clearly.*

References

[NIST-SP800-57-pt1-r5] Barker, Elaine, *NIST Special Publication 800-57 Part 1 Revision 5 Recommendation for Key Management: Part 1 – General*, NIST Special Publication 800-57pt1r5, 2020, URL: https://doi.org/10.6028/NIST.SP.800-57pt1r5.

[NIST-SP800-90A] Barker, Elaine and Kelsey, John, *NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST Special Publication 800-90A, 2015, DOI: 10.6028/NIST.SP.800-90Ar1, URL: http://dx.doi.org/10.6028/NIST.SP.800-90Ar1.

[PSD2] Council of European Union, *Directive (EU) 2015/2366 - Payment services (PSD 2)*, 2015, URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32015L2366.

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

[BCP-212] Denniss, W and Bradley, J, *OAuth 2.0 for Native Apps*, BCP 212, IETF, Oct. 2017, DOI: 10.17487/RFC8252, URL: https://www.rfc-editor.org/info/rfc8252.

[1] Department of the Treasury, *Competition and Consumer (Consumer Data Right) Rules 2020*, 2022, URL: https://www.legislation.gov.au/Details/F2022C00187.

[TDIF-05] Digital Transformation Agency, *05 Role Requirements*, Trusted Digital Identity Framework Release 4.7, Commonwealth of Australia (Digital Transformation Agency), June 2022, URL: https://www.digitalidentity.gov.au/sites/default/files/2022-06/TDIF%2005%20Role%20Requirements%20-%20Release%204.7%20%28Doc%20Version%201.10%29.pdf.

[TDIF-05A] Digital Transformation Agency, *05A Role Guidance*, Trusted Digital Identity Framework Release 4.7, Commonwealth of Australia (Digital Transformation Agency), June 2022, URL: https://www.digitalidentity.gov.au/sites/default/files/2022-04/TDIF%2005A%20Role%20Guidance%20-%20Release%204.6%20%28Doc%20Version%201.5%29.pdf.

[TDIF-ACR-1.3] Digital Transformation Agency, *Authentication Credential Requirements*, Trusted Digital Identity Framework, Commonwealth of Australia (Digital Transformation Agency), 2018, URL: https://dta-www-drupal-20180130215411153400000001.s3.ap-southeast-2.amazonaws.com/s3fs-public/files/digital-identity/Trusted%20digital%20identity%20framework%202/Authentication%20Credential%20Requirements.pdf.

[ISM] Directorate, Australian Signals, *Information Security Manual*, tech. rep., 2022, URL: https://www.cyber.gov.au/sites/default/files/2022-06/Information%20Security%20Manual%20%28June%202022%29.pdf.

[FAPI-2.0-SEC] Fett, D, *FAPI 2.0 Security Profile*, tech. rep., 2022, URL: https://openid.bitbucket.io/fapi/fapi-2_0-security.html.

[2] Fett, D., *FAPI 2.0 Attacker Model*, tech. rep., 2021, URL: https://openid.net/specs/fapi-2_0-attacker-model-01.html.

[3] 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.

[4] 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.

[OAUTH-DCR] Jones, M et al., *OAuth 2.0 Dynamic Client Registration Protocol*, RFC 7591, IETF, July 2015, URL: https://datatracker.ietf.org/doc/html/rfc7591.

[5] Li, Wanpeng and Mitchell, Chris J, "Security issues in OAuth 2.0 SSO implementations", *International Conference on Information Security*, Springer, 2014, pp. 529–541.

[6] 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.

[7] Li, Wanpeng and Mitchell, Chris J., "Analysing the Security of Google's Implementation of OpenID Connect", *Detection of Intrusions and Malware, and Vulnerability Assessment*, ed. by Juan Caballero, Urko Zurutuza, and Ricardo J. Rodríguez, Cham: Springer International Publishing, 2016, pp. 357–376, ISBN: 978-3-319-40667-1, DOI: 10.1007/978-3-319-40667-1_18.

[8] Lodderstedt, T et al., *OAuth 2.0 Security Best Current Practice*, BCP, IETF, Dec. 2021, URL: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics.

[9] 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.

[10] Mladenov, Vladislav, Mainka, Christian, and Schwenk, Jörg, *On the security of modern Single Sign-On Protocols: Second-Order Vulnerabilities in OpenID Connect*, 2016, arXiv: 1508.04324 [cs.CR].

[11] Montjoye, Yves-Alexandre de et al., "Unique in the shopping mall: On the reidentifiability of credit card metadata", *Science* 347.6221 (2015), pp. 536–539, DOI: 10.1126/science.1256297, URL: https://www.science.org/doi/abs/10.1126/science.1256297.

[NIST-SP800-63B] Paul A. Grassi, P et al., *NIST Special Publication 800-63B Digital Identity Guidelines: Authentication and Lifecycle Management*, NIST Special Publication 800-63B, 2017, URL: https://pages.nist.gov/800-63-3/sp800-63b.html.

[FAPI-1.0-BL] Sakimura, N, Bradley, J, and Jay, E, *Financial-grade API Security Profile 1.0 - Part 1: Baseline*, tech. rep., 2021, URL: https://openid.net/specs/openid-financial-api-part-1-1_0.html.

[FAPI-1.0-ADV] Sakimura, N, Bradley, J, and Jay, E, *Financial-grade API Security Profile 1.0 - Part 2: Advanced*, tech. rep., 2021, URL: https://openid.net/specs/openid-financial-api-part-2-1_0.html.

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

[CDR] Treasury Data Standards Body, Commonwealth Department of, *Consumer Data Standards, v1.17.0*, tech. rep., 2022, URL: https://consumerdatastandardsaustralia.github.io/standards.

A CDR Standards covered in this analysis

In preparing this report, we read the following sections of the Data Standards:

- Consumer Experience (all)
- Security Profile (all)
- Dynamic Client Registration APIs (all)
- Register APIs (all)