Analysis of a Decentralised Digital Token Architecture for Public Transport

Oscar King and Geoffrey Goodell

University College London, Gower St, London WC1E 6BT, United Kingdom

Abstract. *Background.* Digitisation is often viewed as beneficial to a user. Where originally people would physically have to identify to a service, pay for a ticket in cash, or go into a library to access a book, people can now achieve all of this through a click of a button. While these actions may seem functionally identical to their analogue counterparts, they come with one important difference. Namely, in the digital case, a user's actions are automatically recorded. The recording of user's interactions presents a problem because this information can be used outside the control of the person whom it concerns. This issue is only exacerbated by the centralisation of these aforementioned services' authentication mechanisms permitting the collection of even more data.
Aim. This work aims to motivate the need and establish the feasibility for the application of a privacy-enhancing digital token management service to public transit.
Method. A proof-of-concept implementation of the Decentralised Digital Identity Architecture proposed by Goodell and Aste is developed. This implementation was optimised for the public transport use case. Finally, its performance is tested in a local environment to better understand the technical challenges and assess such a system's technical feasibility in a production setting.
Results. It was observed that for loads between 1 and 5 requests per second the proof-of-concept performs within acceptable limits with a maximum median response time of 334 milliseconds. Above 5 requests per second response times drastically increase due to hardware bottlenecks.
Conclusions. It was concluded that the demonstrated throughput and latency shows that the system can feasibly compete with solutions currently in use. Yet, further work is needed to demonstrate these performance characteristics in an environment similar to that experienced in production.

Keywords: Digital tokens · Anonymous credentials · Public transport.

1 Introduction

Each day, millions of users interact with digital services. User authentication is often needed to gain access to the service. Users likely trust that during this authentication process their personal information is processed privately and securely. Also, users seek swift, well laid out, and clear interfaces. Users can become quickly frustrated or distrust services that are difficult to use, ultimately they may avoid a service entirely if the usability is a sufficient obstacle to their primary goal [1].

Current widely-used Identity Management Systems (IDMS) such as *GOV.UK Verify* and the *Federal Cloud Credential Exchange* (FCCX) have demonstrated shortcomings related to privacy and security [2]. The increasing number of services adopting federated or centralised authentication systems as well as their increasing user bases only further stresses these privacy and security related issues [3]. These systems introduce a central hub in which it is possible to aggregate and link a user's interactions with different service providers, in addition to having visibility over identifiable information. Unfortunately, in the case of a malicious compromise or a malicious hub operator, it is then possible to undetectably impersonate, profile, and target users [2], leading to an erosion of public trust [4] in the service. Unfortunately, issues with privacy and security are not limited to government-run services. Recently reported vulnerabilities or data leaks in Facebook and Google are examples [5, 6].

Responding to these issues, research has been undertaken in privacy-enhancing identity and credential management systems [7, 8, 9, 10, 11]. These systems individually address a subset of concerns, have their privacy properties, their own merits, and their drawbacks [7]. Currently, the interplay between the specific merits and limitations of these forms of systems in the context of concrete use cases are not well understood. Different use-cases have different design requirements due to the specific needs of the system's stakeholders. Careful analysis is therefore needed to detail the specific considerations system implementers and users need to understand and address. Developers also need to study whether current privacy-enhancing identity and credential management systems can feasibly compete with existing conventional commercial systems.

This report addresses the question: "Given a specific use case, how do privacy-enhancing IDMS compare to the existing solution with respect to its design requirements?". To limit the scope of this work, the question will be explored in the context of the protocol described by Goodell et al [7]. This report refers to the Goodell work as *the protocol* or the *Distributed Digital Identity Architecture* (DigID).

The article continues as follows. The next section provides context for such a system and helps motivates their utility. Additionally, it introduces terminology and possible competing solutions. Section 3 describes the challenges in applying DigID to Transport for London (TfL) and suggests possible extensions to address these shortcomings. Section 4 details the design specific choices in the implementation. Next, the Section 5 outlines the testing approach and results. Finally Sections 6 and 6.1, draw conclusions and evaluate the study, respectively.

2 Background

This section places this work in the current regulatory and economic climate. It also outlines the key concepts underpinning this project as well as relevant previous and similar work completed.

2.1 Societal Context

The last decade has seen significant modernisation of large-scale publicly funded digital services such as that of Transport for London (TfL). Modernisations include i) contact-less payments in London in 2014 [12], ii) the roll-out of OMNY by the Metropolitan Transportation Authority in New York in 2019 [13, 14], and iii) the projected adoption of electronic payment methods for public transit in the Parisian metropolitan area by 2021 [15]. It is argued that large institutions are capable of implementing change only when afforded the right incentives; otherwise, the above-mentioned modernisation might not have taken place. In the case of contactless payments for public transport, modernisation makes the service easier to use and likely increases usage [16]. Where transit is government-run, profit maximisation is not the main goal. Instead, these service operators are typically looking to increase consumer surplus and social good. Therefore increased public usage can be leveraged to decrease price, as well as generate additional revenue to allow further investment to improve services. By extension of this argument, the introduction of contactless payment by these institutions should aid in the adoption and implementation of the DigID protocol, given the right economic incentives. At least two economic justifications support the DigID protocol:

- *Changing public attitude.* People are becoming more aware of what data are being collected and the value of such data [17, 18]. Consumers in the European Union are even willing to forego savings on products and services to preserve rights afforded to them under the General Data Protection Regulation (GDPR) [19]. This willingness to forego savings supports the claim that data privacy can be a significant hurdle when deciding whether to use a service. It is shown that reducing these hassle costs[1] can increase revenue and user retention [16].
- *Potential cost savings.* The reasons for this are twofold. First, depending on the use-case, the costs and risk of maintaining specialised infrastructure can be high. Service providers may wish to outsource the management of system sub-components that are not part of their core businesses. Service provides have previously taken these steps, such as the operation of OMNY and TfL by third parties [20, 21]. Second, the operational costs of the protocol could be much lower than current alternatives. In 2016, TfL awarded Barclaycard a ten-year contract worth up to £380 million as its merchant acquirer [22]. Using a cost model developed by Ernst and Young [23], a similar task can likely be completed with substantial savings.

Similar to the public's attitude towards data privacy, regulation concerning personal data and data privacy has been extended and has become more stringent [24]. The GDPR stipulates how, and for what reasons, personal data can be processed and how it should be kept; with substantive fines being levied against firms or institutions that are found to be non-compliant [25, 26].

[1] Not to be confused with inconvenience.

Limiting the amount of personal data a system can or needs to collect by virtue of its design may be a worthwhile endeavour for the system operators - this is distinct from designing a system that does not hold data after use. The former system could limit the operator's exposure to the risks of collecting personal data. Mitigating these risks may result in the decreased likelihood of fines as well as the possible improved public image and trust. In the case of a third party non-publicly funded the decreased risk of fines is likely the stronger motivator because fines directly decrease companies' profits. Yet for a public institution, the increased trust and improved public image are likely to be of primary concern. As opposed to the former system, the latter system does not enjoy these benefits as it relies on trust that this data is indeed deleted and therefore there is still an inherent risk of data misuse.

2.2 Distributed Ledger Technology and Associated Terminology

In literature, blockchains are occasionally used interchangeably with Distributed Ledger Technology (DLT), they are however distinct concepts. Distributed ledger technology refers to the network of distinct participants, the storage mechanism, as well as the consensus algorithm, blockchains are the underlying data-structure with which many DLT systems are built.

Distributed ledgers can take many different forms and support multiple purposes requiring different combinations of access control. The terminology pertaining to ledger access control and use is not consistent throughout literature. The definitions given below correspond to a common interpretation. There are two dimensions along which DLTs are typically classified in the context of access control. Along one axis a distinction is made between *public*, and *private* DLTs. Along the other the distinction is made between *permissioned* and *permissionless* DLTs. The terms *public* and *private* refer to entities involved in the use of the DLT and the benefit derived from its use. *Public* ledgers allow anyone to benefit either directly or indirectly from its use. *Private* ledgers only provide utility for a fixed consortium of participants. *Permissioned* and *permissionless* refers to the manner by which entities are allowed to directly access the DLT to either write to it or read from it. *Permissioned* DLTs restrict access to an authenticated group of participants. *Permissionless* DLTs do not restrict access in this manner and are open to full participation by anyone who wishes.

The combination of these configurations may provide desirable network properties. Public-permissioned distributed ledgers may be used by governmental agencies such that citizens can access information but cannot alter this ledger's state. Private-permissioned DLTs allow a group of mutually distrusting participants to achieve some common goal, e.g. two businesses exchanging documents, or performing services [27]. The protocol described by Goodell et al. [7] makes use of a DLT system that can be used by anyone but is presumed to be operated by registered and regulated entities. Thus, this report, therefore, limits the scope of discussion to public-permissioned DLTs.

2.3 Competing Privacy-Enhancing Identity Management Systems

There are many different implementations of privacy-enhancing IDMSs, each with their own set of privacy characteristics. For the authors it was important for the protocol to minimise the control points that may be used to compromise the system. This way users need minimal trust that the system is secure. Second, eliminating vectors that could lead to mass surveillance was also deemed fundamental, not least so that users are more likely to feel free and behave naturally, which in turn could improve adoption of the system. Finally, we deemed it fundamental that the system allows users to manage their own data linkages within the system. If this is not the case, then institutions might still be able to profile people based on their actions within the system. The choice was made to use the DigID protocol because we are certain that it was designed with these concerns in mind as is evident from the fundamental design constraints introduced by Goodell et al [7]. We make no claims that these three characteristics are not met by other competing systems.

3 Adapting the Architecture for Transport for London

Transport for London is the governing body regulating all public transport in the London Metropolitan Area (LMA) apart from National Rail services [28]; it governs one of the world's most extensive public transit networks [29]. It owns several subsidiaries that control or operate the public transit services provided in the LMA such as bus and tram services. This report refers to TfL in the general sense to include it and any of its subsidiaries.

Although TfL operates the London Underground system, many other aspects of the network such as selling Oyster credit and the operation of London Buses are contracted out to independent third parties [30]. Much of the required operational infrastructure concerning system participants may already be in place as the system is already partially decentralised. It may, therefore, seem that many of the roles that are needed to operate DigID are therefore already clearly defined. Unfortunately, this is not the case due to, for example, the fact that credit is centrally verified.

The challenges associated with mapping protocol participants onto existing and potential stakeholder and the potential protocol and legislative adaptations that might need to occur are outside the scope of this work. Rather, this section focuses on how the existing protocol would need to be extended so that users can interact with the protocol similarly to the current TfL system. Deviating too far from how users commonly interact with the system may introduce additional barriers, reducing the adoption of the system.

In the simplest problem setting, the user will interact with the system at three points: (a) to purchase Oyster credit, (b) to gain entry to the system after proving sufficient credit, and (c) upon exit of the system to terminate the journey. This problem setting describes what is commonly referred to by TfL as "Pay-as-you-go" (PAYG). It is important to note that both DigID and Oyster

Table 1. Notation used in the subsequent figures depicting protocol extensions. This notation is an extension to the notation used in the original protocol specification found in the paper by Goodell and Aste [7].

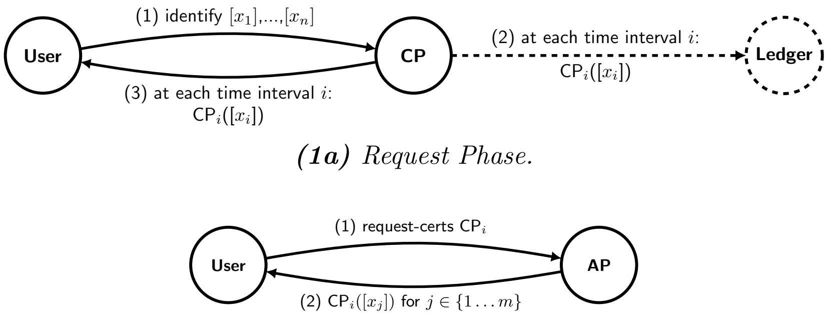
rebate x	A request for a rebate for unspent tokens, with parameter x.
finish x	An attempt to identify to the receiving party, with parameter x, to end an existing session.
escrow x	A request to escrow the credentials x.
finalise x	A request to finalise the transaction of x, after being put into escrow.

cards are not payment schemes, rather tokens and credit on DigID and Oyster cards represent that the user already has paid through a different medium.

In this problem setting there are two main differences between the way a user would interact with the standard DigID protocol and the TfL fare system. First, PAYG calculates journey cost based on entry and exit stations. The DigID protocol only models a single interaction with an Authenticating Party (AP) and Service. There is no way for the protocol to know *a priori* where a user entering the system is going. Similarly, it is impossible for the protocol to know *ex ante* where a user exiting the system came from. Second, if using an Oyster card or payment card, PAYG uses no additional data. This differs from DigID where users are required to download a block of signatures to retrieve their signature. Leaving these differences unaddressed would significantly alter how a user would need to interact with the TfL system and is not desired.

3.1 Protocol Extensions

To allow users to benefit from the network while allowing them to independently configure the amount of data they need or want to process - thereby addressing the second issue raised above, the protocol could be adapted as shown in Figure 1. The difference between the adaptation shown in Figure 1 and the original protocol is that the CPs send the blindly signed credential back to the user (3). This credential can then be presented to an AP to authenticate with a Service. The main reason that the original protocol has the CP publish these blinded credentials to the ledger is that this ensures that the state which the user expects, conforms with the state of the ledger the AP views. Specifically, what a user expects is that the CP is not treating specific users differently such as by using a different public key for each user. By having the AP present blocks from a requested interval to the user for inspection, the user can verify that this is the case because they can inspect a CP's key usage. It must not be the case that a user can request specific keys, because this would still permit the described vulnerability. As opposed to the original protocol, Figure 1 separates the tasks of verifying and token use by giving the User the credential immediately. The user could then request the associated proof block from an AP and verify that the credential received from the CP matches the one received, in addition to


(1) identify $[x_1]$,...,$[x_n]$
User
CP
(2) at each time interval i:
$\mathsf{CP}_i([x_i])$
Ledger
(3) at each time interval i:
$\mathsf{CP}_i([x_i])$
(1) request-certs CP_i
User
AP
(2) $\mathsf{CP}_i([x_j])$ for $j \in \{1 \ldots m\}$

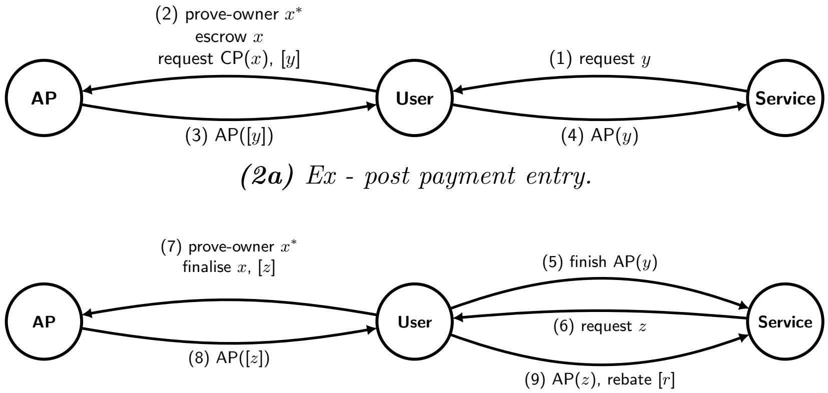
(1a) *Request Phase.*

(1b) *Verify phase.*

Fig. 1. A schematic representation of a variation to the decentralised digital identity architecture that decreases user data usage. In (1a.) the user receives a $\mathrm{CP}_i([x_i])$ at each time interval $i \in \{1 \ldots n\}$ from the CP (3) instead of an arbitrary AP. This $\mathrm{CP}_i([x_i])$ can be used as normal after unblinding, given that CP_i has been published. Diagram (1b) demonstrates how a user would still be able to verify the size of the anonymity set for a given credential. The user would request the proof blocks (1) as normal to a device of their choosing. After receiving them they are free to verify as normal.

the size of the anonymity set. Verification is not, however, a precondition to use. The intention is that users, whenever they wish, could request and verify the blocks associated with their credentials. This could be both before and after use, and report discrepancies if these are found. Research by Heydt-Benjamin et al [31] shows that hybrid systems in which interested users can verify the claimed privacy assurances whilst data-restricted users are free to use the system may provide the similar security and anonymity benefits as systems where all users must verify.

Figure 2 shows the dual-stage approach that could be adopted to address the issue of calculating journey cost as described above. Figure 2a shows the setup or entry phase. The AP uses the ledger to mark the tokens passed in the *escrow* request as used; this request fails if the user has not proven ownership of the token. Only after placing this token into escrow does the AP sign the nonce $[y]$ (2, 3). In the second stage as shown by Figure 2, the user repeats the request for a nonce z from the service. To identify the request, the user must in this stage supply $\mathrm{AP}(y)$ it received from the service in the entry phase (5, 6). The assumption made here is that this is a shared secret. The user then proves ownership of the credential used in the setup phase and requests a signature on $[z]$ (7). The *finalise* request should work similarly to *escrow*; where *escrow* should mark a fresh token as 'in escrow', the *finalise* request should mark a token 'in escrow' as 'spent'. Using $\mathrm{AP}(z)$ the user then requests a rebate which can be calculated based on the nonces y and z. To claim this rebate the user then provides $[r]$ for the service to sign (8). Here *rebate* would work similarly to *request* as used in the original protocol between a User and a CP, and the



(2a) *Ex - post payment entry.*

(2b) *Ex - post payment exit.*

Fig. 2. A schematic representation of an extension to the decentralised digital identity architecture for ex-post variable payments. Figure (2a) demonstrates entry to a given system. The change to the original protocol is the introduction of the *escrow* request. Figure (2b) shows the exit stage - which is new in its entirety. This is achieved by linking this interaction with only that of the interaction presented in Figure (2a) through AP(y).

rebates associated with [r] would be usable after issuance. In both extensions x is modelled as a single token. Yet, it may be necessary to use several tokens to obtain some value greater than that of a single token. This is possible by extending the meaning of x to that of some token vector $\bar{x}$ such that the requests work with several tokens. It is important to note that all operations should work on each element in the vector, not the vector itself.

4 Implementation

To allow for fast and convenient development each system component that serviced requests was written in Python using the Flask micro web framework [32]. Each component was deployed in a docker container. In each container, a gunicorn server ran the component and requests were routed through an nginx proxy server.

Blind signatures are used throughout the protocol and many of the privacy properties the protocol provides rely on effective implementation of these signatures.[2] One common blind signature scheme is that of RSA blind signatures. This scheme has inherent security issues due to the malleability of the ciphertext. Effort was made to find a reputable implementation that avoided these issues, yet

[2] An alternative, valid approach to the design of this protocol involves zero-knowledge proofs; we do not explore that approach here.

none could be found. We did not want to implement our own cryptographic library thus, a Schnorr-based blind signature scheme proposed by Masayuki Abe [33] was used. The implementation of this blinded signature is based on the work of Antonio de la Piedra from the Charm Crypto library [34]. The changes constituted refactoring into classes and fixing type based implementation bugs. Figure 3 details the blinding process devised by Masayuki Abe and segments it into sections referred to in this report and the implementation.

Figure 3 uses notation formally defined below. Definitions have been paraphrased from the original paper:

1. The key generating algorithm $\mathcal{G}$ is a probabilistic polynomial time algorithm defined as $\mathcal{G} : 1^n \to (p, q, g)$ where n is the security parameter and p, q are large primes that satisfy $q|p-1$. Here, g is an element of $\mathbb{Z}_p^*$ of order q, which generates a prime subgroup in $\mathbb{Z}_p^*$ denoted $\langle g \rangle$.
2. Three hash functions are defined: $\mathcal{H}_1 : \{0,1\}^* \to \langle g \rangle$, $\mathcal{H}_2 : \{0,1\}^* \to \langle g \rangle$, and $\mathcal{H}_3 : \{0,1\}^* \to \mathbb{Z}_q$.
3. In order to generate (p, q, g), the key generating algorithm $\mathcal{G} : 1^n$ is executed. Additionally, the signer selects $h \in_U \langle g \rangle$, and secret key $x \in_U \mathbb{Z}_q$. The secret key, together with the generated variables can then be used to compute the real public key y as $y = g^x \bmod p$ and and fixed tag z as $z = \mathcal{H}_1 (p \parallel q \parallel g \parallel h \parallel y)$. If $z = 1$ the algorithm is rerun until $z \neq 1$. The public key is defined as the 6-tuple (p, q, g, h, y, z).

After having run the protocol, the user has signature $\Sigma = (\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$ on message m. This signature can be verified by ensuring that $\zeta \not\equiv 1$ and

$$\varpi + \delta \equiv \mathcal{H}_3 \left(\zeta \parallel \zeta_1 \parallel g^\rho y^\varpi \parallel g^{\sigma_1} \zeta_1^\delta \parallel h^{\sigma_2} \zeta_2^\delta \parallel z^\mu \zeta^\delta \parallel m \right) \bmod q$$

If these conditions hold the user has a valid signature message pair (Σ, m). Readers are referred to the original paper by Masayuki Abe [33] for proofs of correctness and a full protocol specification.

This blind signature scheme is used when interacting with the CP to generate requested credentials, and when interacting with the AP to blind the nonce generated by the service. If following the original protocol specification, the response in the challenge-response (4) is not sent to the user but rather published to the ledger. In the protocol extension proposed, detailed in Figure 1a the challenge-response (4) is sent to both the user as well as published to the ledger alongside other proofs to make up the anonymity set.

The only parameter that was needed to setup the blinding protocol was the security parameter n, this was set to 256. For the use case of TfL, this was deemed sufficient. The security parameter was deemed sufficient for two reasons. First, analysis of Schnorr signatures, on which this scheme is based, suggests that for b bits of security, the group needs to be of size $3b$ and the hash of size $2b$ [35][3]. Second, because there is a maximum monetary value associated with

[3] This proxy is imperfect because the schemes are not identical. Yet, it provides a valuable first estimate.

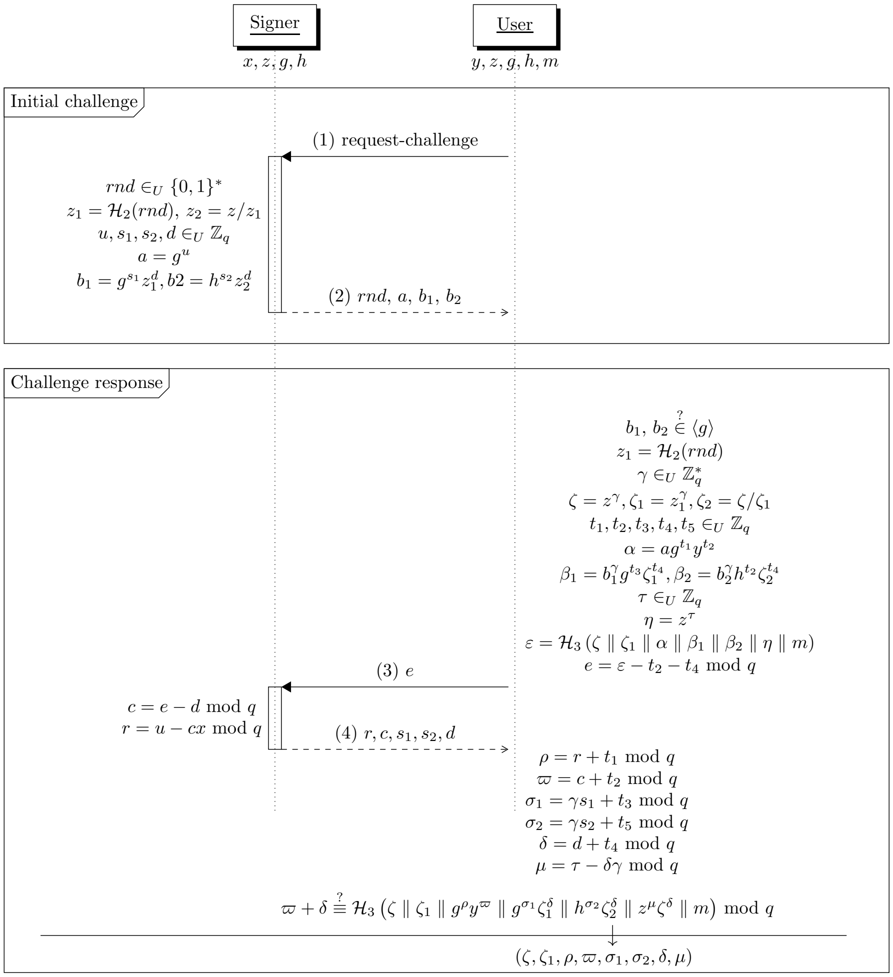

Signer
x, z, g, h
User
y, z, g, h, m
Initial challenge
(1) request-challenge
$rnd \in_U \{0,1\}^*$
$z_1 = \mathcal{H}_2(rnd)$, $z_2 = z/z_1$
$u, s_1, s_2, d \in_U \mathbb{Z}_q$
$a = g^u$
$b_1 = g^{s_1} z_1^d, b2 = h^{s_2} z_2^d$
(2) rnd, a, b_1, b_2
Challenge response
$b_1, b_2 \overset{?}{\in} \langle g \rangle$
$z_1 = \mathcal{H}_2(rnd)$
$\gamma \in_U \mathbb{Z}_q^*$
$\zeta = z^\gamma, \zeta_1 = z_1^\gamma, \zeta_2 = \zeta/\zeta_1$
$t_1, t_2, t_3, t_4, t_5 \in_U \mathbb{Z}_q$
$\alpha = a g^{t_1} y^{t_2}$
$\beta_1 = b_1^\gamma g^{t_3} \zeta_1^{t_4}, \beta_2 = b_2^\gamma h^{t_2} \zeta_2^{t_4}$
$\tau \in_U \mathbb{Z}_q$
$\eta = z^\tau$
$\varepsilon = \mathcal{H}_3 (\zeta \parallel \zeta_1 \parallel \alpha \parallel \beta_1 \parallel \beta_2 \parallel \eta \parallel m)$
$e = \varepsilon - t_2 - t_4 \bmod q$
(3) e
$c = e - d \bmod q$
$r = u - cx \bmod q$
(4) r, c, s_1, s_2, d
$\rho = r + t_1 \bmod q$
$\varpi = c + t_2 \bmod q$
$\sigma_1 = \gamma s_1 + t_3 \bmod q$
$\sigma_2 = \gamma s_2 + t_5 \bmod q$
$\delta = d + t_4 \bmod q$
$\mu = \tau - \delta\gamma \bmod q$
$\varpi + \delta \overset{?}{\equiv} \mathcal{H}_3 \left(\zeta \parallel \zeta_1 \parallel g^\rho y^\varpi \parallel g^{\sigma_1} \zeta_1^\delta \parallel h^{\sigma_2} \zeta_2^\delta \parallel z^\mu \zeta^\delta \parallel m\right) \bmod q$
$(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$

Fig. 3. Schematic representation of the blind signature scheme implemented.The user aborts if any of the checks ($\overset{?}{\in}, \overset{?}{\equiv}$) fail. The user is also assumed to already know the signer's public key (p, q, g, h, y, z). If all checks pass, the signature Σ is the 8-tuple $\Sigma = (\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$. Blinding requires three moves between the signer and the user after the latter initiates a request (1). For each challenge, the signer must issue a response (2). This paper refers to that response as the challenge. The user then sends a challenge-response (3). This challenge-response is subsequently processed and a response is sent that this report refers to as a proof (4). All arithmetic operations are carried out in $\mathbb{Z}_p$ unless stated otherwise.

each block of proofs, if the cost of breaking a key exceeds the combined value of assets in the block it could be argued that there is little motivation for doing so. Therefore, although it is possible to brute force 85 bits of security, it is unlikely that this is economically practical for mass surveillance.

Finally, Hyperledger Fabric was used as the DLT in the implementation. Hyperledger Fabric was chosen because it can process a large number of transactions per second with low latency [27]. This mitigates the risk that the ledger becomes a bottleneck, even at peak times.

5 Results

To compare the performance of the implementation to that of existing production systems it is important to know why a system behaves the way it does, as much as how it performs. This is especially important in the context of system optimisation because understanding why a system performs in a given way allows developers to mitigate potentially unwanted behaviour.

Therefore the tests performed have the following two-part structure. First, each API call is tested, the specifics of which are detailed below. These tests do not mimic the way a user would interact with a system. Rather, they test the performance of the subroutines. Then, a "system test" is performed. Here, the protocol is executed in a way that would resemble a user interaction. Each test is detailed below. Unless stated otherwise each case refers to a request depicted in a figure.

Sub-System Requests

1. *Request nonce:* This corresponds to request (1) in Figure 2a.
2. *Prove ownership:* Request (2) in Figure 2a needed to be split into multiple parts in the implementation. This API call corresponds with the *prove-owner* and *escrow* operations presented.
3. *Request signature:* This to corresponds with the *request* operation shown in request (2) of Figure 2a.
4. *Verify signature:* This test corresponds to request (4) in Figure 2a.
5. *Verify block:* This API call implements Figure 1b in its entirety. Note this request is not on the critical path.

System Requests

6. *Access service:* This is the "system test" that mimics the entire protocol outlined in Figure 2a. To a user, this would be akin to "tapping in" with an Oyster card.

It should be noted that process of requesting credentials has not been included in the testing framework. The time required to process this request scales with the number of tokens requested. It is however possible to do much of the processing outside of the critical path similarly to buying credit on the Oyster card website [36]. The process of obtaining Oyster credit is much less time-bound than the use and validation at a fare gate, which is limited to 500 milliseconds [37] - latencies measured are therefore compared to this 500-millisecond limit. Hypothesised bottlenecks seemed more likely to occur in the use stage of the protocol and it is for this reason the tests focus on this aspect of the protocol.

The implementation tested does not perfectly implement the protocol with extensions established for the TfL use-case in Section 3. Although the specific implementation details differ slightly, the implementation is an accurate proxy for performance. The two main differences are the fact that only the entry stage of the *ex-post payment* mechanism is tested, and that there is no explicit *escrow* request. The current implementation is still an accurate proxy because the escrow command can be dealt with implicitly when proving ownership. No additional API calls need to be made because the AP knows both the intent of the user and has all the primitives that it needs to escrow the credentials. Second, the *ex-post payment* exit stage is not tested. On both entry and exit, blind signatures need to be generated. Where multiple signatures need to be generated, the processing can be safely parallelised[4], such that processing time depends only on the time taken to compute a single signature. Therefore, the *ex-post payment* entryxstage should act as an estimator of the *ex-post payment* exit stage.

Unless specified otherwise the loads used during testing were 1 request per second and 10 requests per second, referred to as the average and maximum load respectively. These loads are derived from the Rolling Origin and Destination Survey (RODS) collected by TfL [38]. This survey presents passenger arrival statistics for each station on the London Underground in 15-minute intervals. The assumption is made that arrivals within this period are uniformly distributed; the reason for this choice being that a uniform distribution is both easy to model and additional information motivating another distribution is lacking. These loads are calculated as follows. First we define the set $\mathcal{I}$ to be the set of all intervals at all stations $\mathcal{S}$. From this we obtain $\mathcal{I}_s \subset \mathcal{I}$, where $s \in \mathcal{S}$ such that $\mathcal{I}_s$ is the set of intervals of station s. An element from this subset can then be obtained by selecting a specific time interval t, such that $i_t \in \mathcal{I}_s$. The average load is calculated by evaluating the following, and is the average of the maximum load across the different stations:

$$load_{avg} = \frac{\sum_{s \in \mathcal{S}} \max_t \mathcal{I}_s}{|\mathcal{S}| \times 60 \times 15}$$

The maximum load is calculated by evaluating the following:

$$load_{max} = \frac{\max_{s \in \mathcal{S}} \left(\max_t \mathcal{I}_s\right)}{60 \times 15}$$

In both cases the result was rounded up to the nearest integer. We divide by 15×60 to transform load per 15 minutes to load per second. Finally, all tests loads were maintained for 20 minutes and run locally on a 3.3 GHz Dual-Core Intel Core i7-6567U CPU. Compared to current widely used mobile CPUs such as the Exynos 8 Octa (8890) or Apple A10 [39, 40], for which the i7-6567U is used as proxy, the i7-6567U is an inferior CPU. The i7-6567U has a lower CPU speed than both mobile CPUs as well as fewer cores that can be used for concurrent computation. It may therefore be the case that the results obtained form a conservative lower bound on what may be feasible.

[4] This is not implemented in the proof of concept but is suggested for a production-grade system.

Table 2. The response-time statistics (in milliseconds) of protocol requests under the average load experienced by the TfL network. The average load the TfL network experiences corresponds to 1 request per second. These loads were maintained for 5 minutes for each test. The distribution statistics are reported on successfully completed requests. Values in bold are those that fall above the 500 millisecond cut-off. S_P denotes Pearson's median skewness.

Requests	Min	Percentiles					Max	σ	S_P
		25	50	75	95	99			
Request nonce	15	23	27	33	54	66	111	11	0.82
Prove ownership	46	71	85	106	177	282	385	43	0.84
Request signature	119	289	394	**520**	**974**	**1308**	**1742**	236	0.70
Verify signature	9	16	19	24	45	61	111	10	0.90
Verify block	113	193	223	275	**635**	**779**	**953**	141	0.98
Access service	75	331	438	**537**	**755**	**926**	**1390**	158	0.34

Table 3. The response sizes in kilobytes of the sub-requests made in Access Service. The Access Service request in its simplest form does not contain the Verify Block request and is therefore not shown. The table does not include data sent in the request.

Request	Response Size (kB)
Request Nonce	0.67
Prove Owner	1.79
Request Signature	5.20
Verify Signature	0.15
Total	7.81

5.1 Test Results

The results presented aim to provide a sufficient description of the response time distributions for the test cases. They focus on the average cases for successful requests. MasterCard limits the transaction time at ticket gates to 500 milliseconds and achieve an average of 300 milliseconds [37]. In order to compare performance with the current system we will use 500 milliseconds as a cut-off for acceptable latency.

5.2 Analysis

This subsection discusses the results obtained in the experiments and analyses the specific patterns observed. Possible solutions are suggested for all the shortcomings discussed. The main takeaway is that although request latency occasionally exceeded 500 milliseconds, the bottlenecks are known and solutions are both known and feasible to implement. We also briefly analyse the possible costs of the proposed solution.

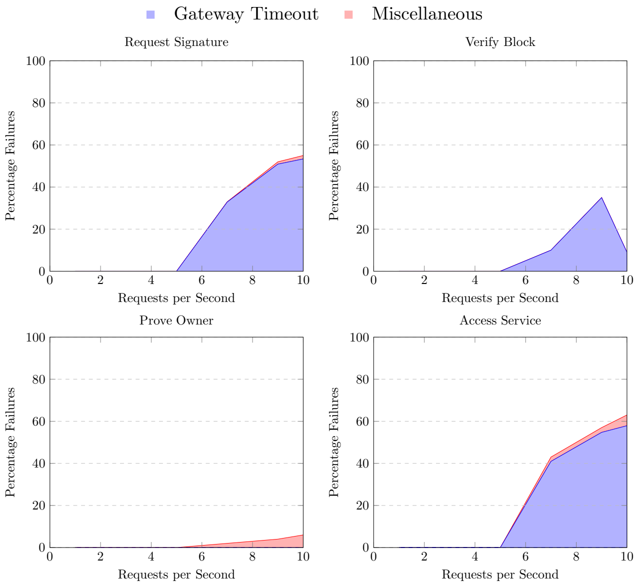


Fig. 4. Percentage of failed requests under different loads decomposed into failure types. Miscellaneous errors include 502 Bad Gateway errors due to server restarts, and 500 Internal Server errors due to database read/write errors.

Analysis of Results At high levels of concurrent users, Figure 5 shows that the system responds to fewer than the number of requests made per second. Fewer requests are processed than received per second which creates a queue, which increases latency. Additionally, many requests fail due to a 60-second gateway timeout. Two approaches are suggested to mitigate this issue and reduce the latency to acceptable levels as experienced at lower loads. First, the number of parallel workers available can be increased such that the load can be balanced between them. Second, the protocol can be further optimised to reduce request processing time. In this case, a single worker will be able to process requests in less time such that throughput is increased. Given the small throughput deficit, re-implementing mathematically heavy tight loops in C/C++ or moving the blinding protocol onto an FPGA could lead to improved performance sufficient for the needs of TfL. Fast implementations of cryptographic algorithms like ECC exist [41], and is a common practice in the field of cryptocurrency mining.

In addition, at higher levels of concurrent users, the requests shown in Figure 4 suffered from a number of failed requests reporting either HTTP Status 500 or 502 codes. This was an issue with the test environment rather than the protocol. The 502 status codes were the result of the gunicorn server restarting before sending a response to the nginx proxy server. The 500 status codes are seemingly caused by the database occasionally, but non-deterministically returning a null value, despite the value having been written to the database.

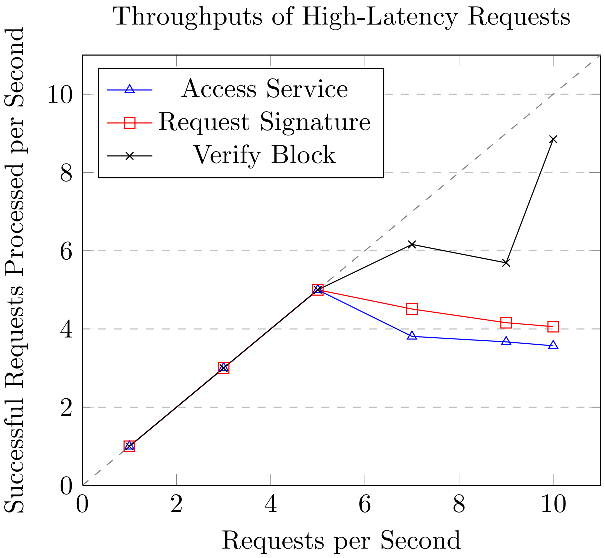


Fig. 5. The throughput of the three test cases for which the measured median latency was over 500 milliseconds in the maximum load scenario. The figure shows the difference in requests made and requests successfully processed under varying loads. The dashed diagonal line shows the point at which the number of incoming requests equals the number of processed requests. Results under this diagonal indicate queue formation. Note, *Verify Block* is not on the critical path.

The fact that the protocol does not meet requirements under maximum load is an issue, but it may not compromise feasibility in most cases. Figure 5 shows that, given the current configuration, users will start experiencing unacceptable latency if there are more than five concurrent users per second. According to the RODS dataset [38] for all stations between 5 am and midnight this load only occurs 0.63% of the time and all occurrences are contained to nine stations. Each of these stations has numerous ticket gates. Groups of these ticket gates could be run on separate hardware and be treated as 'sub-stations'. The correct configuration would decrease the number of concurrent users on each system and reduce load to acceptable levels.

In the *Access Service* request, and each of its subroutines, the use of the ledger has been marginalised out by caching the necessary queries at regular intervals. This is beneficial because it improves user experience by reducing latency. This design choice ledger has its challenges. If not correctly configured the system could permit a double-spend on exit because the result of the *finalise* method part of the *ex-post payment exit* protocol as described in Figure 2 has not been made public. An example of a concrete exploit of this vulnerability would be as follows. A group of people have entered the TfL network through one of the stations without fare gates. Only one individual in the group 'taps in'. Upon

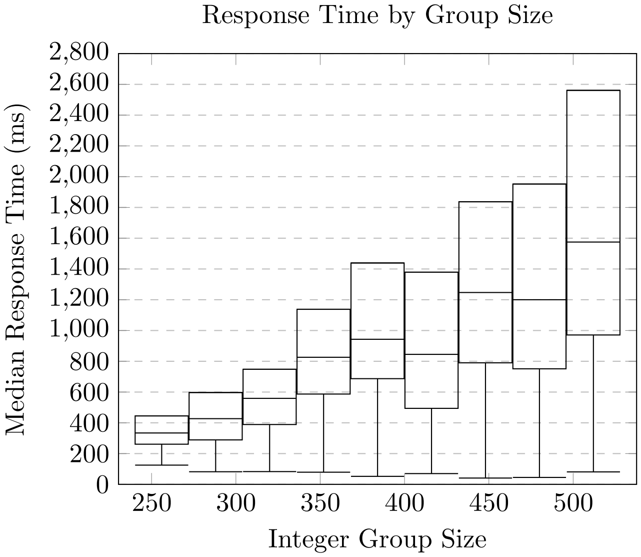


Fig. 6. The median response time for the *Access Service* request. The figure describes the relationship between request latency and the blind signature protocol's security parameter. The top whiskers are omitted for readability however for completeness the maximum values are as follows: 1555, 1884, 3688, 4139, 8513, 5061, 8853, 10909 and, 13770. All in milliseconds.

exit, the individual with the validated credential first 'taps out' by interacting with AP_1. They and pass the device with the credential to the next person in the group who also 'taps out', this time with AP_2. This process is then repeated for different members of the group and APs. Because the APs haven't published the *finalise* request to the ledger, the user can double-spend the tokens. Although it is up to the discretion of the implementing authority, based on their analysis of risks and costs, how they wish to solve this issue, two possible approaches are given. One approach to mitigating this issue would be to reduce the publication period of AP transaction data to a point where it becomes physically impractical to carry out this attack, however, the underlying problem would remain. Another approach would be to locally cache spent signatures and deny exit to double-spenders by checking the local cache. It is important to consider the overall computational cost of introducing additional logic into the protocol. The results presented are likely to be invariant to single small checks, yet if too many are introduced, the results might not be representative and additional testing might be needed.

The *Request Signature* subroutine was the sub-component that contributed most to the latency of the *Access Service* request. It was hypothesised that the blind signing process contributed heavily to this latency, and the larger the group size the longer the execution time. To test this hypothesis, two experiments were

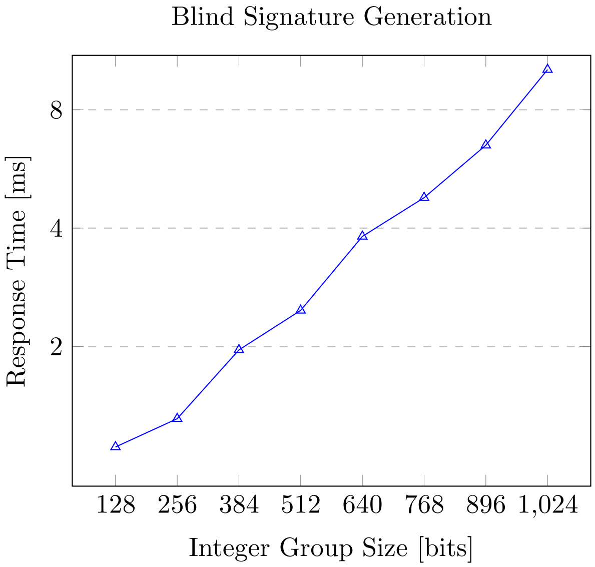


Fig. 7. Time taken to generate a single blind signature. The figure shows how the execution time of a blind signing process varies depending on the size of the integer group in which the arithmetic takes place. This experiment ignores networking overhead and focuses on the execution speed of the cryptographic functions. The range of values on the x-axis was deemed sufficient to characterise both the relationship shown, and to depict the signing times for probable choices in group size ranging from low-security (128 bits) to high-security (1024 bits).

run, the results of which are shown in Figure 6 and Figure 7. The hypothesis that the signing functions contribute heavily to the *Request Signature* latency was not substantiated by the evidence. The minimum response time of *Request Signature* was 112 milliseconds while the execution time of the blind signing process takes 1.31 milliseconds with a group size of 256 bits. The slow response time is likely due to the combination of two factors. First, *Request Signature* has comparatively larger request bodies that need to be marshalled. Second, there are additional network requests that are needed to set up the blind signatures. Results shown in Figure 6 support the former reason. Here, the only variable is the security parameter of the blinding protocol. A higher security parameter leads to larger signatures. Although the request bodies can be optimised further by removing redundancy[5], it is unlikely that much can be done in reducing the fixed networking costs.

[5] The current implementation encodes a value together with its modulus. The protocol that is being used is known *a-priori* and these moduli only need to be defined once. Removing these moduli from subsequent requests would reduce the size of the request body and could speed up marshalling and un-marshalling.

Table 3 depicts the response sizes of the individual sub-requests in the *Access Service* interaction. These values were collected alongside the load data and therefore do not include the outgoing request sizes. The responses, however, are generally larger than the request sizes. Therefore, a conservative upper bound for the *Access Service* interaction can be estimated by doubling the total value. These values were generated based on requesting a signature with a single public key. In the TfL use-case, multiple public keys might need to be escrowed, which would result in a linear increase in request and response sizes. The credit associated with each key and the corresponding ease of use will, therefore, need to be traded off against data usage.

Cost Analysis TfL supports both payment with Oyster Card, which it manages, as well as through contactless card and phone-based mechanisms that are managed externally. In order to feasibly serve the same demographic DigID needs to be able to be run from both a mobile phone as well as a card or smart device issued by TfL - similarly to an Oyster card.

The case for feasibility on mobile phones has already been briefly explored in this section. Given a user is running DigID on a phone there should be no further costs aside from buying credit. We therefore focus on the cost feasibility of a smart device issued by TfL.

The estimating the exact cost of manufacturing is outside the scope of this work. Instead we use the costs of the components we expect to be vital for such a device as a proxy for feasibility. The main components of such a device we expect to be an Radio Frequency (RF) Transceiver, a micro-controller, an FPGA, a battery, and a container. If we assume these items are needed in bulk, such as a couple thousand units we can see that the total unit price is under £10 without including wiring and manufacturing costs[6].

6 Conclusions

To best address the question: “Given a specific use case, how does a privacy-enhancing IDMS compare to the existing solution with respect to its design requirements?” it is decomposed into the following sub-questions:

1. Despite the design-specific compromises made, does the proposed system achieve its primary function while maintaining the core constraints around which the protocol is designed?
2. Given this implementation, what hardware infrastructure is needed to meet the requirements?
3. How does the proposed system compare to the current system with respect to data privacy?

[6] RF Transceiver: £1.53 per unit [42]. Micro-controller: £5.70 per unit [43]. FPGA: £0.892 per unit [44]. Battery: £0.473 per unit [45]. Container: £0.854 per unit [46]

Addressing the first sub-question, the results show that for an average throughput, namely 1 user per second, *Access Service* can process this load in under 500 milliseconds, with the median response time being 438 milliseconds. In this setting DigID can compete with existing solutions. However, in the maximum throughput setting the same conclusion cannot be drawn. More generally, observing the minimum latency for *Access Service* of 75 milliseconds, using the appropriate hardware infrastructure for a given throughput, the protocol can, in its current state, perform as TfL expects current solutions to perform. It can be said that given the appropriate hardware the proof-of-concept demonstrates that a system following the protocol can be used for fare-collection with performance comparable the current fare-collection system.

It may also be concluded that a workable solution is possible using common hardware. At a service level, the performance of the protocol is unlikely to be hardware constrained and should be able to comfortably meet demand in the TfL use case. The identified bottleneck of signature generation only becomes challenging at the infrequent extremes, for which additional hardware can be employed to run the protocol in parallel on separate systems. Finally, given the memory requirements of the protocol, it is possible to conclude that the protocol can be run on mobile devices. Currently, it is not possible to say whether the data usage would be reasonable because it depends on variable factors that need to be set by the implementer such as token value and a more detailed analysis of what could be considered 'reasonable usage'. Given the variable parameters, however, the usage could be calculated in advance and left to the discretion of the user.

If implemented correctly DigID would allow a user to travel identically to that when using the Oyster card. Yet, unlike using the Oyster network, none of the user's trips can be linked. This offers greater privacy because users cannot be profiled based on travel habits. Additionally, payment information, is completely separate from travel habits and the two should be impossible to be reconciled.

6.1 Evaluation

The experimental setup is viewed as a useful 'first step' in discerning the viability of the protocol for the TfL use case. Yet, there are some important limitations to highlight:

- We assume a security parameter of $n = 256$ is sufficient for the blinding protocol. This a compromise between network latency and security. Although it is possible to un-blind the credentials by brute force, it isn't deemed useful economically. Ideally, the security parameter would be increased to $n = 512$ to remove to the possibility of advanced persistent threats un-blinding credentials but this needs to be paired with the message size optimisations described.
- All the experiments were run locally, on a single device. This can be regarded as a strength because it provides a more resource-constrained environment compared to deployment on a separate server. Therefore a protocol that can

meet the requirements in this environment, is more likely to work well in production. It does, however, have a few weaknesses. First, because everything is run locally the results do not take network latency into account. It, not possible to comment to what extent this affects feasibility. Yet, we know that the use of the ledger can be marginalised out in hot loops and state can be written asynchronously. Therefore the main assumption is that the local network serves as a proxy for Near Field Connections. Second, this protocol will be run on phones or other smart devices. Although we substantiate the argument that the current results form a useful proxy, this is still an inference. Both these limitations can be tested and their impact assessed with further investigation. Finally, due to the fact that all experiments were run locally, our ability to repeat experiments was limited for longer running experiments. This is because we would have to run each experiment sequentially which quickly becomes infeasible. This repetition is desired in order to create confidence intervals around results depicted in Figures 4 and 5.

Certain parts of the extensions were not implemented. These extensions include the exit stage of the *ex-post payment* extension. For the *ex-post payment* extension a proxy was used. Although this provides an estimate of this request's performance, it may not capture specific implementation details that significantly affect performance.

6.2 Future Work

Although this project goes some of the way to answering the question of viability in a production setting, it notes its shortcomings and areas that need to be further explored before DigID can be deployed into production.

First, the evaluation noted that the experiments were run locally which may affect system performance and subsequent conclusions are drawn. It is recommended that future work include redeploying the system on multiple nodes to simulate CPs, APs, and Services, as well as running the User code on mobile devices. Results can then be compared against those gathered in this work.

Second, it was mentioned that the proof-of-concept does not perfectly implement the protocol description specified in Section 3. The implementation should be extended such that it implements the complete *ex-post payment* extension. Additionally, it should be investigated to what extent the messages can be optimised. This would permit the analysis of a system that more closely resembles the production system and is more likely to meet TfL requirements in all cases.

Finally, it cannot be expected that all users have equal access to the internet nor can internet access be guaranteed in underground stations. The DigID protocol specifies an offline mode in which the protocol can be run. The implementation should be extended such that both online and offline use is possible. After which the design trade-offs and the performance of the system can be reassessed. All these recommendations are considered executable and would further prove the robustness of the protocol.

References

[1] Sacha Brostoff et al. "Federated identity to access e-government services - Are citizens ready for this?" In: *Proceedings of the ACM Conference on Computer and Communications Security*. 2013, pp. 97–107. ISBN: 9781450324939. DOI: 10.1145/2517881.2517893.

[2] Luís T A N Brandão UL, Nicolas Christin, and George Danezis. "Toward Mending Two Nation-Scale Brokered Identification Systems". In: (2015). DOI: 10.1515/popets-2015-0022.

[3] Tewfiq El Maliki and Jean Marc Seigneur. "A survey of user-centric identity management technologies". In: *Proceedings - The International Conference on Emerging Security Information, Systems, and Technologies, SECURWARE 2007*. 2007, pp. 12–17. ISBN: 0769529895. DOI: 10.1109/SECUREWARE.2007.4385303.

[4] Kieron O'hara. *Transparent Government, Not Transparent Citizens: A Report on Privacy and Transparency for the Cabinet Office*. Tech. rep.

[5] *Facebook Security Breach Exposes Accounts of 50 Million Users - The New York Times*. URL: https://www.nytimes.com/2018/09/28/technology/facebook-hack-data-breach

[6] *New Google+ security bug could affect more than 52 million users, Google says - The Washington Post*. URL: https://www.washingtonpost.com/technology/2018/12/10/googl

[7] Geoffrey Goodell and Tomaso Aste. "A Decentralised Digital Identity Architecture". In: *SSRN Electronic Journal* (Mar. 2019). DOI: 10.2139/ssrn.3342238. arXiv: 1902.08769. URL: https://arxiv.org/pdf/1902.08769.pdf.

[8] Security Team, Research Almaden, and TJ Watson. *Specification of the Identity Mixer Cryptographic Library Version 2.3.0* Specification of the Identity Mixer Cryptographic Library*. Tech. rep. URL: http://domino.watson.ibm.com/library/Cyber

[9] Christian Lundkvist et al. *UPORT: A PLATFORM FOR SELF - SOVEREIGN IDENTITY*. Tech. rep.

[10] Mustafa Al-Bassam et al. "Chainspace: A Sharded Smart Contracts Platform". In: Internet Society, Feb. 2018. DOI: 10.14722/ndss.2018.23241. arXiv: 1708.03778.

[11] Foteini Baldimtsi and Anna Lysyanskaya. "Anonymous credentials light". In: *Proceedings of the ACM Conference on Computer and Communications Security*. 2013, pp. 1087–1098. ISBN: 9781450324779. DOI: 10.1145/2508859.2516687.

[12] *Contactless payments set to launch - Transport for London*. URL: https://tfl.gov.uk/info-for/media/ (visited on 01/29/2020).

[13] *Say hello to tap and go, with OMNY*. URL: https://new.mta.info/system%7B%5C_%7Dmodernization/o (visited on 01/29/2020).

[14] *NYC's contactless subway turnstiles open today with Apple, Google, Samsung and Fitbit Pay support — TechCrunch*. URL: https://techcrunch.com/2019/05/31/nycs-contact (visited on 01/29/2020).

[15] *Ile-de-France : un pas de plus vers la fin du ticket de métro - Le Parisien*. URL: http://www.leparisien.fr/info-paris-ile-de-france-oise/transports/un-pas-de-plus- (visited on 01/29/2020).

[16] Anja Lambrecht and Catherine Tucker. "Paying with Money or with Effort: Pricing When Customers Anticipate Hassle". In: *SSRN Electronic Journal* (Jan. 2012). ISSN: 1556-5068. DOI: 10.2139/ssrn.1805109.

[17] Morning Consult + POLITICO. *National Tracking Poll*. 2019. URL: https://www.politico.com/f/?id=0000016a-700a-dca8-a1ff-7daaa8950001 (visited on 01/30/2020).

[18] Akamai. *Research: Consumer Attitudes Toward Data Privacy Survey, 2018 — Akamai*. Tech. rep.

[19] Moritz Godel, Wouter Landzaat, and James Suter. *Research and analysis to quantify the benefits arising from personal data rights under the GDPR*. Tech. rep. 2017. URL: www.londoneconomics.co.uk.%20https://assets.publishing.service.gov.uk

[20] *Cubic Transportation Systems — OMNY Public Pilot*. URL: https://www.cubic.com/news-events/news (visited on 01/31/2020).

[21] *TfL and Cubic continue partnership - Transport for London*. URL: https://tfl.gov.uk/info-for/media/press-releases/2014/july/tfl-and-cubic-continue-part (visited on 01/31/2020).

[22] Transport for London. *Merchant Acquiring Contract*. Tech. rep. London. URL: http://content.tfl.gov.uk/board-20161108-item07-merchant-acquiring.pdf.

[23] *Total cost of ownership for blockchain solutions*. Tech. rep.

[24] *General Data Protection Regulation (GDPR) – Official Legal Text*. URL: https://gdpr-info.eu/ (visited on 01/20/2020).

[25] "Intention to fine British Airways £183.39m under GDPR for data breach". In: (2019).

[26] *Google hit with £44m GDPR fine over ads - BBC News*. URL: https://www.bbc.co.uk/news/technolog (visited on 01/31/2020).

[27] Elli Androulaki et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *Proceedings of the 13th EuroSys Conference, EuroSys 2018*. Vol. 2018-Janua. ACM, 2018. ISBN: 9781450355841. DOI: 10.1145/3190508.3190538. arXiv: 1801.10228. URL: https://doi.org/10.1145/3190508.3190538.

[28] *About TfL - Transport for London*. URL: https://tfl.gov.uk/corporate/about-tfl/ (visited on 02/09/2020).

[29] *These are the 10 busiest metros in the world — World Economic Forum*. URL: https://www.weforum.org/agenda/2018/11/these-are-the-world-s-10-busiest-metros/ (visited on 02/09/2020).

[30] *London buses - Transport for London*. URL: https://tfl.gov.uk/corporate/about-tfl/culture-and- (visited on 04/08/2020).

[31] Thomas S. Heydt-Benjamin et al. "Privacy for public transportation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2006. ISBN: 3540687904. DOI: 10.1007/11957454_1.

[32] *Foreword — Flask Documentation (1.1.x)*. URL: https://flask.palletsprojects.com/en/1.1.x/fore (visited on 04/08/2020).

[33] Masayuki Abe. *A Secure Three-move Blind Signature Scheme for Polynomially Many Signatures*. Tech. rep.

[34] *protocol_a01 — Charm-Crypto 0.50 documentation.* URL: https://jhuisi.github.io/charm/charm/sch (visited on 04/07/2020).

[35] Gregory Neven, Nigel P Smart, and Bogdan Warinschi. *Hash Function Requirements for Schnorr Signatures.* Tech. rep.

[36] URL: https://oyster.tfl.gov.uk/oyster/entry.do.

[37] MasterCard. "Contactless payments travel well in London". In: (2018). URL: https://www.mastercard.us/content/dam/public/mastercardcom/na/us/en/governments/ot

[38] Transport for London. *TfL Rolling Origin and Destination Survey.* URL: https://data.london.gov.uk/dataset/tfl-rolling-origin-and-destination-survey (visited on).

[39] URL: //www.samsung.com/semiconductor/minisite/exynos/products/mobileprocessor/exynos-8-

[40] Page Version ID: 969925279. July 2020. URL: https://en.wikipedia.org/w/index.php?title=Apple_A

[41] William N. Chelton and Mohammed Benaissa. "Fast elliptic curve cryptography on FPGA". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 16.2 (Feb. 2008), pp. 198–205. ISSN: 10638210. DOI: 10.1109/TVLSI.2007.912228.

[42] Oct. 2020. URL: https://www.mouser.co.uk/ProductDetail/Semtech/SX1233IMLTRT?qs=%5C%2Fha2p

[43] Oct. 2020. URL: https://www.mouser.co.uk/ProductDetail/Renesas-Electronics/R5F572TKCDFB30

[44] Oct. 2020. URL: https://www.mouser.co.uk/ProductDetail/Lattice/ICE40LP384-SG32?qs=zSw%5C%

[45] Oct. 2020. URL: https://www.mouser.co.uk/ProductDetail/Murata-Electronics/SR626?qs=l7cgNq

[46] Oct. 2020. URL: https://www.mouser.co.uk/ProductDetail/Bud-Industries/PB-1574?qs=W%5C%252