Anonymous Single-Sign-On for n Designated Services with Traceability

Jinguang Han, Liqun Chen, Steve Schneider, Helen Treharne, and Stephan Wesemeyer(✉)

Department of Computer Science, University of Surrey, Guildford, Surrey GU2 7XH, UK
s.wesemeyer@surrey.ac.uk

Abstract. Anonymous Single-Sign-On authentication schemes have been proposed to allow users to access a service protected by a verifier without revealing their identity. This has become more important with the introduction of strong privacy regulations. In this paper we describe a new approach whereby anonymous authentication to different verifiers is achieved via authorisation tags and pseudonyms. The particular innovation of our scheme is that authentication can occur only between a user and its designated verifier for a service, and the verification cannot be performed by any other verifier. The benefit of this authentication approach is that it prevents information leakage of a user's service access information, even if the verifiers for these services collude. Our scheme also supports a trusted third party who is authorised to de-anonymise the user and reveal her whole service access information if required. Furthermore, our scheme is lightweight because it does not rely on attribute or policy-based signature schemes to enable access to multiple services. The scheme's security model is given together with a security proof, an implementation and a performance evaluation.

Keywords: Anonymous Single-Sign-On · Security · Privacy
Anonymity

1 Introduction

Single-Sign-On (SSO) systems are a user-friendly way of allowing users access to multiple services without requiring them to have different usernames or passwords for each service. SSO solutions (e.g. OpenID 2.0 [35] by the OpenID foundation or Massachusetts Institute of Technology (MIT)'s Kerberos [33]) are designed to make the users' identities and possibly additional personal identifiable information (PII) available to the verifiers of the services which they wish to access. However, for some services, a verifier may not require the user's identity (nor any associated PII), just that the user is authorised to access the desired service. Moreover, the introduction of more stringent obligations with regards to

J. Lopez et al. (Eds.): ESORICS 2018, LNCS 11098, pp. 470–490, 2018.
https://doi.org/10.1007/978-3-319-99073-6_23

the handling of PII in various jurisdictions (e.g. GDPR in Europe [20]), requires service providers to minimise the use of PII.

Anonymous Single-Sign-On schemes [19,26,29,38] exist which can protect a user's identity, but may not do so for all entities within a scheme. Moreover, a user's service request can be verified by all verifiers of a system and not just the one it is intended for, which may pose a potential privacy risk to both the user and that verifier. Our proposed scheme addresses these issues and provides the following features: (1) only one authentication ticket is issued to a user, even if she wants to access multiple distinct services; (2) a user can obtain a ticket from a ticket issuer anonymously without releasing anything about her personal identifiable information — in particular, the ticket issuer cannot determine whether two ticket requests are for the same user or two different users; (3) a designated verifier can determine whether a user is authorised to access its service but cannot link different service requests made by the same user nor collude with other verifiers to link a user's service requests; (4) designated verifiers can detect and prevent a user making multiple authentication requests using the same authentication tag ("double spend") but cannot de-anonymise the user as a result; (5) tickets cannot be forged; and (6) given a user's ticket, a central verifier is authorised to recover a user's identity as well as the identities of the verifiers for the requested services in the user's ticket.

Our contributions are: a novel anonymous single-sign-on scheme providing the above features; its associated security model and security definitions; a corresponding formal proof of its security as well as an empirical performance analysis based on a Java-based implementation of our scheme.

1.1 Related Work

We now look at previous research which is most closely related to our scheme in the areas of: (i) Anonymous Single-Sign-On protocols, (ii) anonymous authentication schemes, (iii) multi-coupon schemes and (iv) designated verifiers signature schemes.

Anonymous Single-Sign-On Schemes

One of the anonymous Single-Sign-On system was proposed by Elmufti *et al.* [19] for the Global System for Mobile communication (GSM). In their system, a user generates a different one-time identity each time they would like to access a service and, having authenticated the user, a trusted third party will then authenticate this one-time identity to the service provider. Consequently, the user is anonymous to the service provider but, unlike in our scheme, not the trusted third party who authenticated the one-time identity.

In 2010, Han *et al.* [26] proposed a novel dynamic SSO system which uses a digital signature to guarantee both the unforgeability and the public verification of a user's credential. In order to protect the user's privacy, their scheme uses broadcast encryption which means that only the designated service providers can check the validity of the user's credential. Moreover, zero-knowledge proofs are used to show that the user is the owner of those valid credentials to prevent

impersonation attacks. However, again unlike our scheme, the user is still known to the trusted third party which issued the credentials.

Wang *et al.* [38], on the other hand, propose an anonymous SSO based on group signatures [3]. In order to access a service, the user generates a different signature-based pseudonyms from her credentials and sends the signature to the service provider. If the signature is valid, the service provider grants the user access to the service to the user; otherwise, the service request is denied. The real identities of users can be identified by using the opening technique in [3]. While the user remains anonymous, their scheme (unlike ours) does not, however, provide designated verifiers, i.e. all verifiers can validate a user's request.

Lastly, Lee [29] proposed an efficient anonymous SSO based on Chebyshev Chaotic Maps. In this scheme, an issuer, the "smart card processing center", issues secret keys to users and service providers when they join in the system and to access a service, a user and service provider establish a session key with their respective secret keys. If the session key is generated correctly, the service request is granted; otherwise, it is denied. However, unlike our scheme, each service provider knows the identity of the user accessing their service.

While in [29,38], a user can access any service in the system by using her credentials, in our scheme, a user can only access the services which she selects when obtaining a ticket but can do so while remaining completely anonymous to both issuer and service provider.

Anonymous Authentication Schemes

With respect to anonymous authentication solutions, we consider schemes whose primary feature is to support multiple anonymous authentication. As in our scheme, anonymous authentication enables users to convince verifiers that they are authorised users without releasing their exact identities.

Teranishi *et al.* [37] proposed a k-times anonymous authentication (k-TAA) scheme where the verifiers determine the number of anonymous authentication that can be performed. The k-TAA scheme provides the following two features: (1) no party can identify users who have been authenticated within k times; (2) any party can trace users who have been authenticated more than k times. The verifier generates k tags and for each authentication, a user selects a fresh tag. Nguyen *et al.* [34] proposed a similar dynamic k-TAA scheme to restrict access to services not only the number of times but also other factors such as expiry date.

Camenisch *et al.* [9] proposed a periodic k-TAA scheme which enables users to authenticate themselves to the verifiers no more than k times in a given time period but supports reuse of the k times authentication once the period is up. In this scheme, the issuer decides the number of anonymous authentication request a user can make in a given time period. When a user makes an anonymous authentication request, he proves to a verifier that he has obtained a valid CL signature [11] from the issuer.

Note, however, that our scheme also prevents a verifier from establishing whether a user has used any of the other services thereby also guaranteeing verifier anonymity.

Furthermore, in all of these k-TAA schemes [9,34,37], authentication is not bound to a particular verifier, whereas in our scheme authentication tags are bound to specific verifiers. Moreover, k-TAA schemes allow verifiers to determine a user's identity who has authenticated more than k times while in our scheme multiple authentications to a single verifier is considered "double spending" which a verifier can detect but which does not lead to the de-anonymisation of a user. However, to prevent users from potentially abusing the system, our scheme allows for a central verifier who, given a user's ticket, can extract from it both the user's and verifiers' public keys using the authentication tags contained within it and thus establish the identities of both the user and her associated verifiers.

Lastly, Camenisch *et al.* in [13] and the IBM identity mixer description of its features in [27] define a scheme that has similar properties to ours including that of a central verifier (called "inspector") trusted to reveal a user's identity. The scheme is based on users obtaining a list of certified attributes from an issuer and the users using a subset of their attributes to authenticate to verifiers. The distinguishing difference between their scheme and ours is that their verification of anonymous credentials is not bound to a designated verifier whereas our is.

Multi-coupon Schemes

There is some degree of similarity between our scheme and a number of multi-coupon schemes. Armknecht *et al.* [1] proposed a multi-coupon scheme for federated environments where multiple vendors exist. In [1], a user can redeem multiple coupons anonymously with different vendors in an arbitrary order. To prevent double-spending of a coupon, a central database is required to record the transaction of each multi-coupon. The main difference to our scheme is that each coupon can be redeemed against any service provider while our authentication tags can only be validated by its designated verifier. Moreover, our "double-spend" detection is done by the verifier and does not require a central database.

Similarly, the schemes propose by Liu *et al.* [31] which provides strong user privacy and where a user can use an e-coupon anonymously no more than k times before his identity can be recovered. However, the user's coupons can be redeemed against any service rather than a designated verifier as our scheme provides.

Designated Verifiers

Jakobsson in [28] introduced the concept of a designated verifier which means that in a proof we ascertain that nobody but this verifier can be convinced by that proof while the authors in [21] present an anonymous attribute-based scheme using designated-verifiers. In their work they focus on identifying multiple designated verifiers. This is achieved through using the verifier's private key in the verification so that no other third party can validate the designated verifier signature. We adopt the high level concept of a designated verifier in our approach, i.e. given a valid authentication tag for service A, only service A's verifier can establish its validity. As this property is conceptually similar to

the designated signatures described in [21,28], our verifiers are called designated verifiers. However, this is where the similarity ends with Jakobsson's designated verifiers. Notably, in [28], a verifier cannot convince others that the signature is from the signer because the verifier can generate the signature by himself. In our scheme, everyone can check that the authentication tags are signatures generated by the ticket issuer.

In summary, while a number of previous authentication schemes address the anonymity of the user and multiple authentications, the novelty of our work is that we ensure no information leakage across verifiers, since authentication can only occur between a user and its designated verifier while also providing a central verifier who can de-anonymise the user and reveal the identity of the verifiers in case of a misbehaving user. To the best of our knowledge, our anonymous Single-Sign-On scheme using designated verifiers is the first which has been formally presented in term of definitions, security models and proven to be secure under various cryptographic complexity assumptions together with an empirical performance evaluation.

1.2 Paper Organisation

This paper is organised as follows: Sect. 2 provides a high-level overview of the scheme and its claimed security properties; Sect. 3 outlines the applicable security model; Sect. 4 introduces the cryptographic building blocks and notation used throughout this paper; Sect. 5 describes the formal construction of our while Sect. 6 presents the theorems for its security proof; Sect. 7 provides a performance evaluation of our scheme; and Sect. 8 concludes the paper with directions for future work. The full version of this paper in [25] contains detailed formal definitions, security models and proofs of the scheme.

2 Scheme Overview and Security Properties

Entities in Our Proposed Scheme

Before providing a high-level overview of our anonymous single-sign-on scheme, we first introduce the various entities in the scheme as shown in Fig. 1, and define their purpose and roles: the **Central Authority ($\mathcal{CA}$)** is a trusted third party responsible for establishing the cryptographic keys and parameters used in the scheme and issues credentials to the other entities in the scheme; a **User ($\mathcal{U}$)** is someone who wishes to access some distinct services anonymously; the **Ticket Issuer ($\mathcal{I}$)** issues tickets to registered, yet anonymous users for the requested services; a **Designated Verifier ($\mathcal{V}$)** is a verifier for a specific service that a user might want to access; the **Central Verifier ($\mathcal{CV}$)** is another trusted third party which is allowed to retrieve the identities of the user, $\mathcal{U}$, and the verifiers, $\mathcal{V}$s, from the authentication tags present in a user's ticket, T_U; an **Authentication Tag (Tag_V)** is both tied to a user, $\mathcal{U}$, and a designated verifier, $\mathcal{V}$ and is used to prove to the designated verifier that the user is a valid user and allowed to

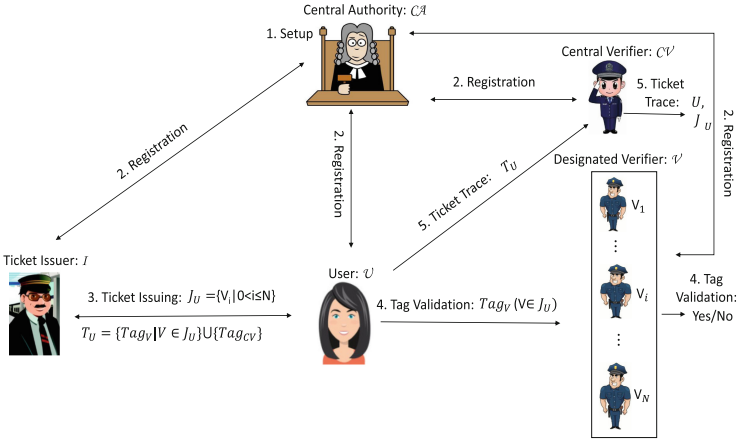

Fig. 1. Interaction of the various entities in our scheme

access the associated service; a **Ticket** (T_U) contains the authentication tags for the services a user, $\mathcal{U}$, has requested.

Overview of Proposed Scheme

Figure 1 illustrates at a high-level how our scheme works. For its detailed formal construction, please refer to Sect. 5. Conceptually, our scheme operates as follows: **Registration:** The issuer, verifiers, central verifier and users all register with the CA. **Ticket Issuing:** A user decides which services (and thus which verifiers) she wants to access and requests an appropriate ticket from the issuer. **Tag Validation:** To access a service, the user presents the appropriate authentication tag to the service. The validity period and any other restrictions of the tag can be captured in the free text part of the tag or be a default set by the verifier. If a user's tag is valid then the user is logged in to the service. Note that, unlike some other Single-Sign-On systems, the issuer does not need to be on-line for the tag validation to succeed. **"Double-Spend" detection:** If the user present the same tag twice then the verifier can warn the user that she is already logged in and that she should resume the already existing session or offer to terminate the previous session and start with a fresh one. **Ticket trace:** If a user is seen to abuse the service (e.g. violate the terms and conditions), the central verifier might be called upon to de-anonymise the user and determine any other services she has used.

Security Properties in Our Proposed Scheme

Having defined the different entities and described how they interact, we now list the security properties of our scheme:

– **User Anonymity:** In our scheme, users use pseudonyms whenever they interact with the issuer or a verifier. As such, the issuer cannot link a user

across different ticket requests. Similarly, a user's identity is also hidden from a designated verifier.
- **Authentication Tag Unlinkability:** Apart from the central verifier and the issuer, no set of colluding verifiers can establish whether two or more different authentication tags came from the same anonymous user.
- **Verifier Anonymity:** The verifier's identify is protected from other users and verifiers, i.e. given an authentication tag, only the designated verifier can validate it and no other verifier (apart from the central verifier and the issuer) can determine for whom it is.
- **Designated Verifiability:** Given an authentication tag, Tag_V for verifier, $\mathcal{V}$, only $\mathcal{V}$ can validate it.
- **"Double-spend" detection:** Any verifier, $\mathcal{V}$, can detect when a user attempts to re-use an authentication tag but cannot de-anonymise the user.
- **Unforgeability:** Neither tickets nor individual authentication tags can be forged by any colluding users or verifiers.
- **Traceability:** There exists a trusted third party, a central verifier, who can, given a user's ticket, T_U, retrieve the user's and the verifiers' public keys (and hence their respective identities) from the authentication tags contained within T_U.

In the next section, we provide the security models in which these properties hold while Sect. 6 contains the associated theorems which are used to prove those models.

3 Security Model Overview

We now present a high-level overview of the security models which are used to prove the security of our scheme. The models are defined by the following games executed between a challenger and an adversary. Detailed formal security models as well as their proofs are presented in the full version of this paper [25] which also demonstrates the correctness of our scheme.

Unlinkability Game

This game covers the security properties of user anonymity, authentication tag unlinkability, verifier anonymity, designated verifiability and "double spend" detection. In this game verifiers and other users can collude but cannot profile a user's whole service information. In other words, no party can link different tags to the same user and determine a verifier's identity included in an authentication tag (thus proving verifier anonymity) except for the designated verifier, the ticket issuer or the central verifier. Moreover, for each authentication tag, the adversary can query its validity once, which in the context of this game addresses the properties of designated verifiability and "double spending".

Unforgeability Game

This game focuses on proving the unforgeability property of our scheme. Users, verifiers and the central verifier can collude but cannot forge a ticket on behalf of the ticket issuer.

Table 1. Syntax summary

Syntax	Explanations	Syntax	Explanations
1^ℓ	A security parameter	V_i	The i-th ticket verifier
$\mathcal{CA}$	Central authority	J_U	The service set of $\mathcal{U}$ consisting of the identities of ticket verifiers & ID_{CV}
$\mathcal{I}$	Ticket issuer		
$\mathcal{V}$	Ticket verifier	PP	Public parameters
$\mathcal{U}$	User	Ps_U	A set of pseudonyms of $\mathcal{U}$
$\mathcal{CV}$	Central verifier	Ps_V	The pseudonym generated for $\mathcal{V}$
ID_I	The identity of $\mathcal{I}$	Tag_V	An authentication tag for $\mathcal{V}$
ID_V	The identity of $\mathcal{V}$	Tag_{CV}	An authentication tag for $\mathcal{CV}$
ID_U	The identity of $\mathcal{U}$	T_U	A ticket issued to $\mathcal{U}$
ID_{CV}	The identity of $\mathcal{CV}$	$\|X\|$	The cardinality of the set X
$\epsilon(\ell)$	A negligible function in ℓ	$x \xleftarrow{R} X$	x is randomly selected from the set X
σ_I	The credential of $\mathcal{I}$	$A(x) \rightarrow y$	y is computed by running the algorithm $A(\cdot)$ with input x
σ_V	The credential of $\mathcal{V}$		
σ_U	The credential of $\mathcal{U}$	$\mathcal{KG}(1^\ell)$	A secret-public key pair generation algorithm
σ_{CV}	The credential of $\mathcal{CV}$		
MSK	Master Secret Key	$\mathcal{BG}(1^\ell)$	A bilinear group generation algorithm
H_1, H_2	Cryptographic hash functions	p, q	Prime numbers

Traceability Game

This game focuses on the traceability property of our scheme. It shows that even if users, verifiers and the central verifier collude, they cannot generate a ticket which is linked to a user who has never obtained a ticket or a user who is not the real owner of the ticket.

4 Preliminaries

In this section, we introduce the cryptographic building blocks used by our scheme including bilinear groups, the BBS+ signature scheme, zero knowledge proofs and various complexity assumptions needed to ensure its security. The mathematical notation and symbols used throughout this paper are summarised in Table 1.

4.1 Bilinear Groups and Pairings

In our scheme, bilinear groups are used to support the BBS+ signature scheme (defined in Sect. 4.2 below).

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\tau$ be three cyclic groups with prime order p. A pairing is defined to be a bilinear, non-degenerative and computable map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow$

$\mathbb{G}_\tau$[7]. Given a security parameter, 1^ℓ, we define $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ to be a bilinear group generation algorithm. Note that Galbraith, Paterson and Smart [22] classified parings into three basic types and our scheme is based on the Type-III pairing where the elements on $\mathbb{G}_1$ are short (≈160 bits). This was chosen because for all $g \in \mathbb{G}_1$ and $\mathfrak{g} \in \mathbb{G}_2$, there exists an polynomial-time efficient algorithm to compute $e(g, \mathfrak{g}) \in \mathbb{G}_\tau$ resulting in a more efficient algorithm.

4.2 BBS+ Signature

Based on the group signature scheme [6], Au, Susilo and Mu [2] proposed the BBS+ signature. This signature scheme works as follows:

- Setup: Let $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, h be a generator of $\mathbb{G}_1$ and $g, g_0, g_1, \cdots, g_n$ be generators of $\mathbb{G}_2$.
- KeyGen: The signer selects $x \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $Y = h^x$. The secret-public key pair is (x, Y).
- Signing: To sign a block message $(m_1, m_2, \cdots, m_n) \in \mathbb{Z}_p^n$, the signer selects $w, e \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $\sigma = (g_0 g^w \prod_{i=1}^n g_i^{m_i})^{\frac{1}{x+e}}$. This signature on $(m_1, m_2, \cdots, m_n)$ is (w, e, σ).
- Verification: Given a signature (w, e, σ) and $(m_1, m_2, \cdots, m_n)$, the verifier checks $e(Yh^e, \sigma) \overset{?}{=} e(h, g_0 g^w \prod_{i=1}^n g_i^{m_i})$. If so, the signature is valid; otherwise, it is invalid.

Au, Susilo and Mu [2] reduced the security of the above signature to the q-SDH assumption (see Definition 2 below) in Type-II paring. Recently, Camenisch, Drijvers and Lehmann [8] reduced its security to the JOC-version-q-SDH assumption (see Definition 3 below) for Type-III pairing.

4.3 Zero-Knowledge Proof

In our scheme, zero-knowledge proof of knowledge protocols are used to prove knowledge and statements about various discrete logarithms including: (1) proof of knowledge of a discrete logarithm modulo a prime number [36]; (2) proof of knowledge of equality of representation [15]; (3) proof of knowledge of a commitment related to the product of two other commitments [12]. We follow the definition introduced by Camenish and Stadler in [14] which was formalised by Camenish, Kiayias and Yung in [10]. By PoK:$\{(\alpha, \beta, \gamma) : \Upsilon = g^\alpha h^\beta \wedge \tilde{\Upsilon} = \tilde{g}^\alpha \tilde{h}^\gamma\}$, proof on knowledge of integers α β and γ such that $\Upsilon = g^\alpha h^\beta$ and $\tilde{\Upsilon} = \tilde{g}^\alpha \tilde{h}^\beta$ hold on the groups $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\tilde{\mathbb{G}} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$, respectively. The convention is that the letters in the parenthesis (α, β, γ) represent the knowledge which is being proven by using the other values to which the verifier can have access.

4.4 Complexity Assumptions

The security of our scheme relies on a number of complexity assumptions defined in this subsection.

Definition 1 (Discrete Logarithm (DL) Assumption [24]). *Let $\mathbb{G}$ be a cyclic group with prime order p and g be a generator of $\mathbb{G}$. Given $Y \in \mathbb{G}$, we say that the discrete logarithm (DL) assumption holds on $\mathbb{G}$ if for all adversary can output a number $x \in \mathbb{Z}_p$ such that $Y = g^x$ with a negligible advantage, namely*

$$Adv_{\mathcal{A}}^{DL} = \Pr\left[Y = g^x | \mathcal{A}(p, g, \mathbb{G}, Y) \rightarrow x\right] \leq \epsilon(\ell).$$

The DL assumption is used in the proof of the traceability property of our scheme.

Definition 2 (*q*-Strong Diffie-Hellman (*q*-SDH) Assumption [4]). *Let $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. Suppose that g and $\mathfrak{g}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Given a $(q+2)$-tuple $(g, g^x, g^{x^2}, \cdots, g^{x^q}, \mathfrak{g}) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2$, we say that q-strong Diffie-Hellman assumption holds on $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ if for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ can output $(c, g^{\frac{1}{x+c}}) \in \mathbb{Z}_p \times \mathbb{G}_1$ with a negligible advantage, namely $Adv_{\mathcal{A}}^{q-SDH} = \Pr[\mathcal{A}(\mathfrak{g}, g, g^x, g^{x^2}, \cdots, g^{x^q}) \rightarrow (c, g^{\frac{1}{x+c}})] \leq \epsilon(\ell)$, where $c \in \mathbb{Z}_p - \{-x\}$.*

Definition 3 ((JOC Version) *q*-Strong Diffie-Hellman (JOC-*q*-SDH) Assumption [5]). *Let $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. Given a $(q+3)$-tuple $(g, g^x, \cdots, g^{x^q}, \mathfrak{g}, \mathfrak{g}^x) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2^2$, we say that the JOC- q-strong Diffie-Hellman assumption holds on the bilinear group $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ can output $(c, g^{\frac{1}{x+c}}) \in \mathbb{Z}_p \times \mathbb{G}_1$ with a negligible advantage, namely $Adv_{\mathcal{A}}^{JOC-q-SDH} = \Pr\left[(c, g^{\frac{1}{x+c}}) \leftarrow \mathcal{A}(g, g^x, \cdots, g^{x^q}, \mathfrak{g}, \mathfrak{g}^x)\right] < \epsilon(\ell)$, where $c \in \mathbb{Z}_p - \{-x\}$.*

The security of the BBS+ signature used in our scheme relies on both the (q-SDH) and JOC-q-SDH) assumptions.

Definition 4 (Decisional Diffie-Hellman (DDH) Assumption [18]). *Let $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. Give a 3-tuple $(\xi, \xi^\alpha, \xi^\beta, T) \in \mathbb{G}_1^3$, we say that the decisional Deffie-Hellman assumption holds on $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ can distinguish $T = \xi^{\alpha\beta}$ or $T = M$ with negligible advantage, namely $Adv_{\mathcal{A}}^{DDH} = |\Pr[\mathcal{A}(\xi, \xi^\alpha, \xi^\beta, T = \xi^{\alpha\beta}) = 1] - \Pr[\mathcal{A}(\xi, \xi^\alpha, \xi^\beta, T = M) = 1]| < \epsilon(\ell)$ where $M \overset{R}{\leftarrow} \mathbb{G}_1$.*

Note that the DDH assumption is believed to be hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$ for the Type-III pairing [23] used in our scheme which means that we actually makes use of the following stronger complexity assumption.

Definition 5 (Symmetric External Diffie-Hellman (SXDH) Assumption [23]). *Let $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. We say that the symmetric external Diffie-Hellman assumption holds on $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ if the decisional Diffie-Hellman (DDH) assumption holds on both $\mathbb{G}_1$ and $\mathbb{G}_2$.*

5 Scheme Construction

In this section, we present a more detailed description of the interactions (cf. Fig. 1) between the entities of our scheme. These interactions are: (i) System Set-up, (ii) Registration, (iii) Ticket Issuing, (iv) Tag Verification and (v) Ticket Tracing. Moreover, we provide details of the mathematical operations involved in these interactions. Formal definitions of the algorithms presented in this section can be found in the full version of this paper [25].

5.1 System Set-Up

Figure 2 shows the details of the system initialisation in which the central authority $\mathcal{CA}$ generates a master secret key, MSK, and the required public parameters, PP. **Note:** Once the system has been set up, all communication between the different entities in our scheme is assumed to be over secure, encrypted channels which can be established by the various entities using standard Public Key Infrastructure. This ensures that our scheme is not susceptible to simple Man-In-The-Middle attacks.

System Set-up:$\mathcal{CA}$ runs $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\tau$. Let $g, h, \xi, \tilde{h}$ be generators of the group $\mathbb{G}_1$ and $\mathfrak{g}$ be generators of $\mathbb{G}_2$. Suppose that $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ are two cryptographic hash functions. $\mathcal{CA}$ selects $x_a \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $Y_A = \mathfrak{g}^{x_a}$. The master secret key is $MSK = x_a$ and the public parameters are $PP = (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau, g, h, \xi, \tilde{h}, \mathfrak{g}, Y_A, H_1, H_2)$.

Fig. 2. System set-up algorithm

5.2 Registration

Figure 3 depicts the registration processes. When registering with the $\mathcal{CA}$, $\mathcal{I}$, $\mathcal{V}$, $\mathcal{U}$ and $\mathcal{CV}$ use the PP and generate their own secret-public key pairs. They then send their identities and associated public keys to $\mathcal{CA}$ which, after receiving a registration request from an entity, uses MSK to generate the corresponding credential for them. Note that only the ticket issuer has two public keys, Y_I and $\tilde{Y}_I$. The first one is used to sign the tickets while the second one is used to validate the ticket.

5.3 Ticket Issuing

During the ticket issuing process (shown in Fig. 4), the user $\mathcal{U}$ defines J_U to be the set containing the identities of the ticket verifiers whose services she wants to access as well as the identity of the central verifier. In order to request a ticket from $\mathcal{I}$, $\mathcal{U}$ creates pseudonyms, (P_V, Q_V), for each $ID_V \in J_U$ by using her secret key to protect the anonymity of the verifiers. She also produces a proof of knowledge of her credentials and submits this proof together with the set J_U and

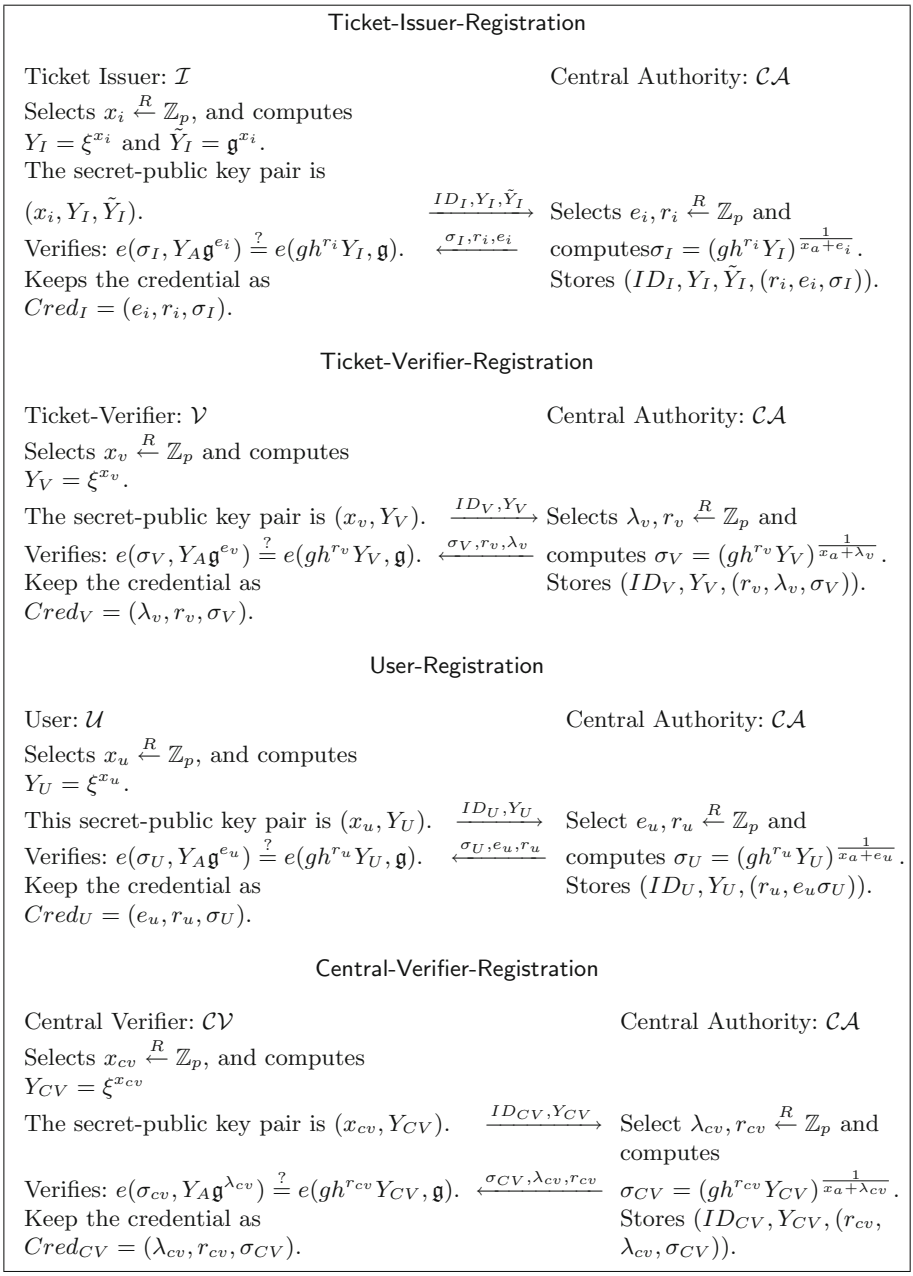

Fig. 3. Registration algorithm

the pseudonyms to $\mathcal{I}$ to convince him that she is a registered user and created the pseudonyms. Once $\mathcal{I}$ has received this information and verified the proof of knowledge, he generates an authentication tag Tag_V for each $ID_V \in J_U$ as well

Ticket-Issuing

Let J_U is $\mathcal{U}$'s list of the identities of verifiers which $\mathcal{U}$ wants to access as well as ID_{CV}

User: $\mathcal{U}$		Ticket Issuer: $\mathcal{I}$
Computes $B_U = gh^{r_u}Y_U$		
Select $v_1, v_2, z_u \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $v_3 = \frac{1}{v_1}$, $\bar{\sigma}_U = \sigma_U^{v_1}$, $v = r_u - v_2 v_3$, $\bar{B}_U = B_U^{v_1} h^{-v_2}$, $\tilde{\sigma}_U = \bar{\sigma}_U^{-e_u} B_U^{v_1} (= \bar{\sigma}_U^{x_a})$, $(z_v = H_1(z_u \Vert ID_V)$, $P_V = Y_U Y_P^{z_v}$, $Q_V = \xi^{z_v})_{ID_V \in J_U}$		
Computes the proof $\prod_U^1$: PoK$\{(x_u, r_u, e_u, \sigma_U, v_1, v_2, v_3, v, (z_v)_{ID_V \in J_U}) : \frac{\tilde{\sigma}_U}{\bar{B}_U} = \bar{\sigma}_U^{-e_u} h^{v_2} \wedge g^{-1} = \bar{B}_U^{-v_3} \xi^{x_u} h^v \wedge (P_V = \xi^{x_u} Y_P^{z_v} \wedge Q_V = \xi^{z_v})_{V \in J_U}\}$ Let $\Delta = (((P_V, Q_V)_{ID_V \in J_U}), \bar{\sigma}_U, \tilde{\sigma}_U, \bar{B}_U, J_U, \prod_U^1)$	$\xrightarrow{\Delta}$	Verifies $\prod_U^1$ and $e(\bar{\sigma}_U, Y) \overset{?}{=} e(\tilde{\sigma}_U, \mathfrak{g})$. Selects $t_u \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $C_U = \xi^{t_u}$ For $ID_V \in J_U$, selects $d_v, w_v, e_v \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $D_V = H_2(C_U \Vert ID_V)$, $E_V = \xi^{d_v}$, $F_V = Y_V^{d_v}$, $K_V = Y_V Y_P^{d_v}$, $s_v = H_1(P_V \Vert Q_V \Vert E_V \Vert F_V \Vert K_V \Vert Text^a)$ and $Z_V = (gh^{w_v} \tilde{h}^{s_v})^{\frac{1}{x_i + e_v}}$ Let $Tag_V = (P_V, Q_V, E_V, F_V, K_V, Text, s_v, w_v, e_v, Z_V)$ For the central verifier ID_{CV}, selects $w_{cv}, e_{cv} \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $s_{cv} = H_1(s_1 \Vert s_2 \Vert \cdots \Vert s_{\vert J_U \vert})$ and $Z_{CV} = (gh^{w_{cv}} \tilde{h}^{s_{cv}})^{\frac{1}{x_i + e_{cv}}}$
a) For $ID_V \in J_U$, verify $D_V \overset{?}{=} H_2(C_U \Vert ID_V)$, $s_v \overset{?}{=} H_1(P_V \Vert Q_V \Vert E_V \Vert F_V \Vert K_V \Vert Text)$ and $e(Z_V, \tilde{Y}_I \mathfrak{g}^{e_v}) \overset{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$. b) Verify $s_{cv} \overset{?}{=} H_1(s_1 \Vert s_2 \Vert \cdots \Vert s_{\vert J_U \vert})$ and $e(Z_{CV}, \tilde{Y}_I \mathfrak{g}^{e_{cv}}) = e(gh^{w_{cv}} \tilde{h}^{s_{cv}}, \mathfrak{g})$ c) Keep (z_u, C_U) secret	$\underset{C_U}{\overset{T_U}{\leftarrow}}$	The ticket is: $T_U = \{(D_V, Tag_V) \vert V \in J_U\} \cup \{s_{cv}, w_{cv}, e_{cv}, Z_{CV}\}$ where s_v and s_{cv} are the serial numbers of Tag_V and T_U, respectively.

[a] *Text* consists of the system version information and all other information which can be used by verifiers to validate the tag, e.g. valid period, tag type, *etc.*

Fig. 4. Ticket issuing algorithm

as an overall Tag_{CV} for $\mathcal{CV}$ in case the ticket needs to be traced. Note that these tags are constructed using the public keys of the respective verifiers and thus can only be validated by the corresponding $\mathcal{V}$ or the central verifier, $\mathcal{CV}$. The ticket is formed from these individual tags. Note that each tag and the overall ticket are signed by the issuer using his private key while the integrity of the

tags and the overall ticket is assured using hashes of their respective content. The ticket is sent back to $\mathcal{U}$ who verifies the integrity of each tag and the overall ticket using the supplied hash values as well as that each tag and the overall ticket have been signed by the issuer.

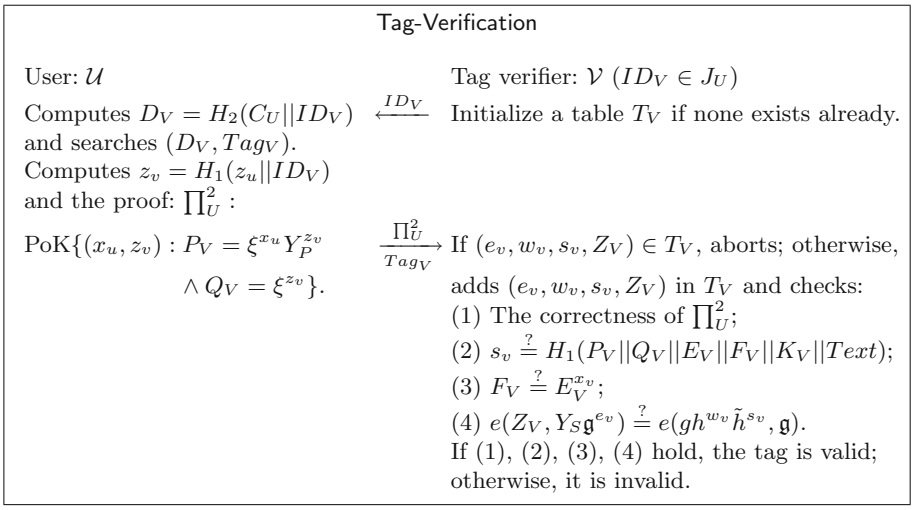

Fig. 5. Tag Verification algorithm

5.4 Tag Verification

The tag verification process is shown in Fig. 5. When the user $\mathcal{U}$ wants to access a service, the ticket verifier $\mathcal{V}$ send his identity information to the user which $\mathcal{U}$ uses to look up the corresponding tag, Tag_V. In order to access the service, $\mathcal{U}$ must submit a proof of knowledge of her secret key alongside the relevant authentication tag Tag_V to prevent users from sharing authentication tags. $\mathcal{V}$ checks his table of previously received tags to ensure that the tag has not already been used previously (double-spend detection), before verifying the user's proof of knowledge in Step 1. Step 2 checks the integrity of the tag using a hash function while Step 4 verifies that it has been issued by the ticket issuer, $\mathcal{I}$. Step 3 can only be verified by $\mathcal{V}$ as it requires the private key of the verifier. Only if $\mathcal{V}$ can complete all steps successfully, is the user granted access.

5.5 Ticket Tracing

Lastly, in the case that a user $\mathcal{U}$'s whole service information J_U needs to be traced, the central verifier, $\mathcal{CV}$, sends its identity to $\mathcal{U}$ who is then required to submit the information, Π_U^2, Tag_{CV}, (which is the same information as that of the Tag Verification algorithm) as well as her overall ticket. Note that, provided a

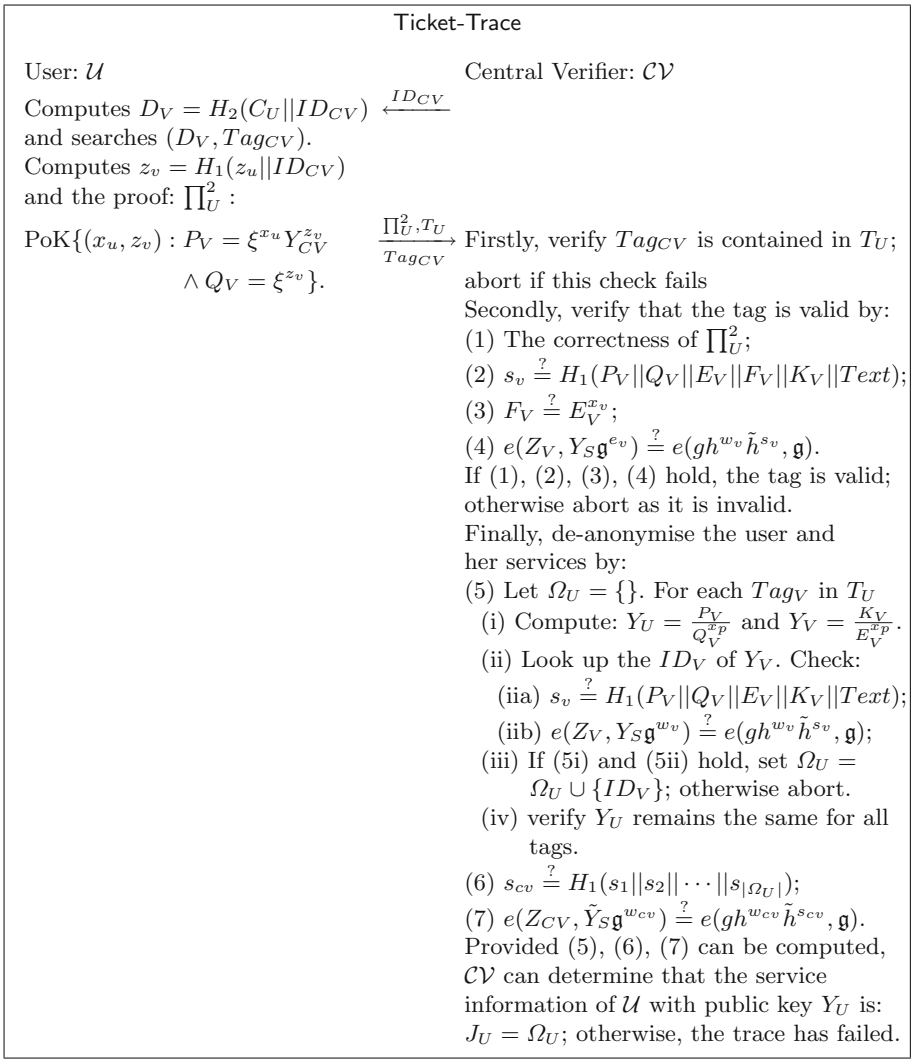

Fig. 6. Ticket trace algorithm

single tag is known, the whole ticket information could also be obtained directly from the issuer, $\mathcal{I}$, in case the user is not co-operating.

On receipt of this information, the central verifier first validates that the submitted tag Tag_{CV} passes the standard verification process (see Sect. 5.4) as the central verifier's ID_{CV} is always included in J_U. As discussed previously, this steps ensures that $\mathcal{U}$ is a valid user and that the tag belongs to her. Once this steps has passed, the central verifier can then validate the integrity of the ticket and that the previously presented authentication tag is indeed part of

the ticket which establishes that the ticket does indeed belong to the user who presented it. Using his private key, the central verifier can now compute the user $\mathcal{U}$'s public key as well as the public keys of all the verifiers contained within the authentication tags and thus determine the user's identity and her service information J_U.

6 Security Analysis

In this section we present the theorems which establish the security of our scheme. Their proofs can be found in the full version of this paper [25].

Theorem 1 (Unlinkability). *An anonymous Single-Sign-On for n designated services with traceability scheme in Figs. 2, 3, 4, 5 and 6 is* $(\boldsymbol{\rho_1, \rho_2, \rho_3, \epsilon'(\ell)})$**-selectively unlinkable** *if the DDH assumption holds on the bilinear group* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *with the advantage at most* $\epsilon(\ell)$*, and* H_1, H_2 *are secure cryptographic hash functions, where* ϱ_1 *is the total number of verifiers selected by* $\mathcal{A}$ *to query tickets,* ϱ_2 *is the number of ticket validation queries,* ϱ_3 *is the number of ticket trace queries,* $\epsilon(\ell) = \frac{\epsilon'(\ell)}{2}$.

Theorem 2 (Unforgeability). *An anonymous Single-Sign-On for n designated services with traceability scheme in Figs. 2, 3, 4, 5 and 6 is* $(\boldsymbol{\varrho, \epsilon'(\ell)})$**-unforgeable** *if the JOC-version-q-SDH assumption holds on the bilinear group* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *with the advantage at most* $\epsilon(\ell)$*, and* H_1, H_2 *are secure cryptographic hash functions, where* ϱ *is the total number of verifiers selected by* $\mathcal{A}$ *to query tickets,* $\varrho \leq q$*,* $\epsilon(\ell) = (\frac{p-q}{p} + \frac{1}{p} + \frac{p-1}{p^3})\epsilon'(\ell)$.

Theorem 3 (Traceability). *An anonymous Single-Sign-On for n designated services with traceability scheme in Figs. 2, 3, 4, 5 and 6 is* $(\boldsymbol{\rho, \epsilon(\ell)})$**-traceable** *if the q-SDH assumption holds on the bilinear group* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *with the advantage at most* $\epsilon_1(\ell)$*, the DL assumption holds on the group* $\mathbb{G}_1$ *with the advantage at most* $\epsilon_2(\ell)$*, and* H_1, H_2 *are secure cryptographic hash functions, where* $\epsilon(\ell) = max\left\{\frac{\epsilon_1(\ell)}{2}(\frac{p-q}{p} + \frac{1}{p} + \frac{p-1}{p^3}), \frac{\epsilon_2(\ell)}{2}\right\}$*,* ϱ *is the total number of ticket issuing queries made by* $\mathcal{A}$ *and* $\varrho < q$.

7 Benchmarking Results

In order to evaluate the performance of our scheme, it has been implemented in Java using a benchmarking framework [17] to extract the computational timings of the algorithms. The benchmark was executed on a Dell Inspiron Latitude E5270 laptop with an Intel Core i7-6600U CPU, 1TB SSD and 16 GB of RAM running Fedora 27. Our implementation makes use of bilinear maps using elliptic curves as well as other cryptographic primitives. The implementation of the scheme relies on the JPBC library [16] for the bilinear maps and uses the cryptographic functions provided by bouncycastle [30]. Note that the Java based implementation of the JPBC API [16] was used throughout.

Table 2. Benchmark results (in ms)

Protocol phase	Entity	$r = 160$ bits		$r = 320$ bits	
System Initialisation - Central Authority ($\mathcal{CA}$)					
Initialise the system	CA	1398		3385	
Registration - Issuer ($\mathcal{I}$)					
Generate I credentials	CA	12		45	
Verify I credentials	I	641		979	
Registration - User ($\mathcal{U}$)					
Generate user credentials	CA	12		20	
Verify user credentials	User	301		498	
Registration - Central Verifier ($\mathcal{CV}$)					
Generate CV credentials	CA	9		23	
Verify CV credentials	CV	269		497	
Registration - Verifier ($\mathcal{V}$)					
Generate V credentials	CA	10		23	
Verify V credentials	V	290		623	
Tag Verification - Verifier ($\mathcal{V}$)					
Retrieve Tag_V & generate Π_U^2	User	13		34	
Verify Π_U^2 & Tag_V	V	225		575	
Issuing phase					
Protocol phase	Entity	V = #verifiers			
		2	3	2	3
Generate Π_U^1 & ticket request	User	93	101	280	309
Verify Π_U^1, generate ticket	Issuer	481	515	916	1044
Verify ticket	User	764	960	1960	2567
Ticket Tracing - Central Verifier ($\mathcal{CV}$)					
Retrieve ticket T_U & Tag_{CV}; generate Π_U^2	User	8	9	33	37
Verify Π_U^2, Tag_{CV}; trace T_U	CV	983	1146	2575	3182

7.1 Timings

Table 2 shows the results of the computational time spent in the various phases of our proposed scheme which required more complex computations (i.e. some form of verification using bilinear maps or generation of zero knowledge proofs). The bilinear map used in the protocol implementations was a Type F elliptic curve provided by the JPBC library where G is the group of points of the elliptic curve and $|G| = p$ is its prime order whose binary representation requires r-bits. We chose to benchmark primes p with $r = 160$ bits and $r = 320$ bits using 2 or 3 verifiers per ticket. The number of verifiers only impacts on the issuing and ticket tracing phases while the size of r impacts on all phases. The generation of

credentials by the $\mathcal{CA}$ for the issuer, user and the (central) verifiers during the registration phase of the protocol is on average around 12 ms for $r = 160$ bits and 30 ms for $r = 320$ bits while the verification of those credentials by the various parties takes about 300 ms and 650 ms for 160 bits and 320 bits respectively. It can be seen from Table 2 that the current implementation of the our scheme is reasonably fast for elliptic curves when $r = 160$ (e.g. $\approx$1.5 s and $\approx$250 ms for ticket issuing and verification respectively) and still acceptable for $r = 320$ bits ($\approx$4 s and $\approx$600 ms for the same steps). Moreover, it should be possible to improve the performance of the code considerably by pre-computing static values off-line where possible and switching from the current Java-based version to using a Java-wrapper to the C-based implementation of the pbc libraries [32], instead.

8 Conclusion and Future Work

Previous Anonymous Single-Sign-On schemes usually protect the user's identity from other verifiers but not always the issuer nor the verifier to whom the user needs to authenticate. However, previously, the identity of these verifiers has not been considered extensively and neither has the need to ensure that only a designated verifier can validate a given access request. In this paper we proposed an Anonymous Single-Sign-On scheme which enables users and verifiers to remain anonymous throughout while protecting the system from misbehaving users through a central verifier who can, if required, trace the identities of a user and her associated verifiers. Moreover, we provided a formal security model and proofs for the security properties of our scheme as well as an implementation demonstrating the feasibility of deployment.

In our scheme, a user can currently only authenticate to a verifier once as there is only one authentication tag for each verifier in a user's ticket. If the user needs to authenticate herself to a verifier, $\mathcal{V}$, multiple times, she must request additional tickets with the required authentication tag for $\mathcal{V}$ from the issuer. Our scheme could alternatively be amended to allow multiple authentication tags per verifier in each ticket. In this case the scheme's security model and proofs would need to be amended to support this.

Anonymous Single-Sign-On was the main motivational use case for our scheme, but there are other scenarios to which the could be applied, e.g. the purchase of tickets for tourist attractions, where being able to issue a ticket through an Android implementation would be appropriate. Initial results however demonstrate that the timings on an Android client are significantly slower, for example ticket validation can take $\approx$200 times longer than on the laptop. Future work will focus on improving the scheme's performance further (especially on the Android platform) by moving from a pure Java-based implementation to a C-based version as well as performing pre-computations of static values required by proofs of knowledge where possible. Lastly, extending our scheme with an option for users to enable the controlled release of personal information to a given verifier, e.g. by letting a user control which verifier is allowed to de-anonymise her authentication tag, is another area of future research.

Acknowledgement. This work has been supported by the EPSRC Project DICE: "Data to Improve the Customer Experience", EP/N028295/1. The authors would also like to thank the anonymous reviewers and Dr François Dupressoir for their valuable feedback and comments.

References

1. Armknecht, F., Löhr, H., Manulis, M., Sadeghi, A.-R., et al.: Secure multi-coupons for federated environments: privacy-preserving and customer-friendly. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 29–44. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79104-1_3
2. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-TAA. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 111–125. Springer, Heidelberg (2006). https://doi.org/10.1007/11832072_8
3. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38
4. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
5. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptol. **21**(2), 149–177 (2008)
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_3
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
8. Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous attestation using the strong Diffie Hellman assumption revisited. In: Franz, M., Papadimitratos, P. (eds.) Trust 2016. LNCS, vol. 9824, pp. 1–20. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45572-3_1
9. Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., Meyerovich, M.: How to win the clonewars: efficient periodic n-times anonymous authentication. In: ACM CCS 2006, pp. 201–210. ACM (2006)
10. Camenisch, J., Kiayias, A., Yung, M.: On the portability of generalized Schnorr proofs. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 425–442. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_25
11. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_20
12. Camenisch, J., Michels, M.: Proving in zero-knowledge that a number is the product of two safe primes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 107–122. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_8
13. Camenisch, J., Mödersheim, S., Sommer, D.: A formal model of identity mixer. In: Kowalewski, S., Roveri, M. (eds.) FMICS 2010. LNCS, vol. 6371, pp. 198–214. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15898-8_13

14. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052252
15. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_7
16. De Caro, A., Iovino, V.: JPBC: Java pairing based cryptography. In: ISCC 2011, pp. 850–855. IEEE (2011)
17. DICE Project: Benchmark E-ticketing Systems (BETS) (2017). https://github.com/swesemeyer/BenchmarkingETicketingSystems
18. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Inf. Theory Soc. **22**(6), 644–654 (1976)
19. Elmufti, K., Weerasinghe, D., Rajarajan, M., Rakocevic, V.: Anonymous authentication for mobile single sign-on to protect user privacy. Int. J. Mob. Commun. **6**(6), 760–769 (2008)
20. European Commission and European Council: Regulation (EU) 2016/679: General Data Protection Regulation (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN
21. Fan, C.I., Wu, C.N., Chen, W.K., Sun, W.Z.: Attribute-based strong designated-verifier signature scheme. J. Syst. Softw. **85**(4), 944–959 (2012)
22. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discret. Appl. Math. **156**(16), 3113–3121 (2008)
23. Ghadafi, E., Smart, N.P., Warinschi, B.: Groth–Sahai proofs revisited. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 177–192. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_11
24. Gordon, D.M.: Discrete logarithms in GF(P) using the number field sieve. SIAM J. Discret. Math. **6**(1), 124–138 (1993)
25. Han, J., Chen, L., Schneider, S., Treharne, H., Wesemeyer, S.: Anonymous Single-Sign-On for n services with traceability (2018). https://arxiv.org/abs/1804.07201
26. Han, J., Mu, Y., Susilo, W., Yan, J.: A generic construction of dynamic single sign-on with strong security. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICSSITE, vol. 50, pp. 181–198. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16161-2_11
27. IBM Research Zürich: Identity mixer (2018). https://www.zurich.ibm.com/identity_mixer/
28. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_13
29. Lee, T.F.: Provably secure anonymous single-sign-on authentication mechanisms using extended chebyshev chaotic maps for distributed computer networks. IEEE Syst. J. **12**(2), 1499–1505 (2015)
30. Legion of the Bouncy Castle Inc: Bouncy Castle Crypto APIs. https://www.bouncycastle.org/
31. Liu, W., Mu, Y., Yang, G., Yu, Y.: Efficient e-coupon systems with strong user privacy. Telecommun. Syst. **64**(4), 695–708 (2017)
32. Lynn, B.: The pairing-based cryptography (PBC) library (2010). https://crypto.stanford.edu/pbc/
33. MIT Kerberos: Kerberos: The network authentication protocol (2017). https://web.mit.edu/kerberos/

34. Nguyen, L., Safavi-Naini, R.: Dynamic k-times anonymous authentication. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 318–333. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_22
35. Recordon, D., Reed, D.: OpenID 2.0: a platform for user-centric identity management. In: DIM 2006, pp. 11–16. ACM (2006)
36. Schnor, C.P.: Efficient signature generation by smart cards. J. Cryptol. **4**(3), 161–174 (1991)
37. Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication (extended abstract). In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 308–322. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_22
38. Wang, J., Wang, G., Susilo, W.: Anonymous single sign-on schemes transformed from group signatures. In: INCoS 2013, pp. 560–567. IEEE (2013)