Luís T. A. N. Brandão*, Nicolas Christin, George Danezis, and Anonymous

# Toward Mending Two Nation-Scale Brokered Identification Systems

**Abstract:** Available online public/governmental services requiring authentication by citizens have considerably expanded in recent years. This has hindered the usability and security associated with credential management by users and service providers. To address the problem, some countries have proposed nation-scale identification/authentication systems that intend to greatly reduce the burden of credential management, while seemingly offering desirable privacy benefits. In this paper we analyze two such systems: the *Federal Cloud Credential Exchange* (FCCX) in the United States and *GOV.UK Verify* in the United Kingdom, which altogether aim at serving more than a hundred million citizens. Both systems propose a brokered identification architecture, where an online central hub mediates user authentications between identity providers and service providers. We show that both FCCX and GOV.UK Verify suffer from serious privacy and security shortcomings, fail to comply with privacy-preserving guidelines they are meant to follow, and may actually degrade user privacy. Notably, the hub can link interactions of the same user across different service providers and has visibility over private identifiable information of citizens. In case of malicious compromise it is also able to undetectably impersonate users. Within the structural design constraints placed on these nation-scale brokered identification systems, we propose feasible technical solutions to the privacy and security issues we identified. We conclude with a strong recommendation that FCCX and GOV.UK Verify be subject to a more in-depth technical and public review, based on a defined and comprehensive threat model, and adopt adequate structural adjustments.

**Keywords:** NSTIC, IDAP, identification, authentication, surveillance, privacy enhancing technologies, secure two-party computation

DOI 10.1515/popets-2015-0022
Received 2015-02-15; revised 2015-05-12; accepted 2015-05-15.

## 1 Introduction

The realm of services available to citizens via digital online platforms has significantly expanded in recent years. As a result, users and service providers have experienced an increasing burden of managing multiple credentials for identification and authentication. This burden can be reduced with the help of *identity providers*, available on demand to certifiably validate the identity of users.

In this paper, we consider hub-based *brokered identification*, in which an online central entity, called a hub, serves as a broker to mediate the interaction between users, service providers and identity providers. As a broker, the role of the hub is to ensure interoperable identification and authentication, while seemingly offering desirable privacy, security and usability guarantees. Ideally, user privacy benefits from hiding the identity provider and the service provider from one another; the service provider can safely rely on the identity assertions received from the hub about the user; and overall usability improves since the user does not need to ‘remember’ a large number of authentication credentials.

It is very challenging to design a brokered identification scheme that adequately integrates well-needed properties of privacy, security, usability, auditability and forensic capabilities. In this paper, we investigate dangers arising from two national proposals of brokered identification systems, and recommend solutions to repair them.

In the United States, the *National Strategy for Trusted Identities in Cyberspace* (NSTIC) [The11] was published in 2011. NSTIC laments that the “weakness of privacy protections often leave individuals, business and government reluctant to conduct major transactions online,” and thus aims to make “online transactions more secure for business and consumers.” As a national strategy, NSTIC lays the vision, principles and objectives for the development of solutions that will impact how more than a hundred million citizens manage their identity online. Two key components of NSTIC are to leave to users the choice of whom to entrust with their identity, and to allow the private sector to develop the needed identity solutions. In particular, NSTIC sets a basis for the development of a new Identity Ecosystem, outlining the role of different actors, such as the private sector and the government at the federal and other levels.

In the UK, a similar drive exists, spearheaded by the Government Digital Service, part of the Cabinet Office, in the form of an *Identity Assurance Programme* (IDAP). IDAP constituted a “Privacy and Consumer Advisory Group” that since

***Corresponding Author: Luís T. A. N. Brandão:** Carnegie Mellon University & University of Lisbon; luis.papers@gmail.com
**Nicolas Christin:** Carnegie Mellon University; nicolasc@cmu.edu
**George Danezis:** University College London; g.danezis@ucl.ac.uk
**Anonymous:** 06ac01f8898481dd 2acdaacbe7cea1fd 5cdec8e65fe87db5 8605e865b1860f8e (SHA256 commitment)

2012 has refined a set of nine Identity Assurance Principles [Pri14]. For instance, Principle 4 states that “My interactions only use the minimum data necessary to meet my needs” explicitly referring to data processed at identity providers and service providers to fulfill requests “in a secure and auditable manner.” This refers not only to “personal data,” but also to “relationship data” that allows inferring relationship between the user and other providers.

Building on NSTIC and IDAP respectively, the US and the UK have been developing respective nation-scale brokered identification schemes: the *Federal Cloud Credential Exchange* (FCCX), recently rebranded as Connect.GOV, in the US [Uni13]; and the *GOV.UK Verify*, in the UK [Ide13a]. FCCX and GOV.UK Verify share the common goal of solving the problem of identification and authentication in multiple public services, with possible future extensions to the private sector.

It has been acknowledged that “identity services ... will need to align internationally over time” [Wre12]; and indeed, despite certain differences, FCCX and GOV.UK Verify do share striking resemblances. We illustrate their high-level architectures in Fig. 1. An online central *hub* mediates all interactions between service providers (hereafter denoted as *relying parties*, RPs) and private-sector identity providers (IDPs), possibly also involving additional *attribute providers* (ATPs). As a result of an *identification transaction* (links 1–10 in the figure), the relying party identifies and authenticates the user. In GOV.UK Verify a *matching service* (MS) also helps validate assertions from identity providers. Both systems are constrained by arguable structural decisions, such as restricting the *user-agent* (a web browser) to a passive role in the protocol, except for selecting and authenticating to the IDP and relaying messages between other parties.

The FCCX and GOV.UK Verify systems seemingly provide desirable privacy and security properties, but they fail to adequately relate them to the hub. For example, the FCCX solicitation calls for *unlinkability* as a desirable property of hiding the IDP and RP from each other. However, contrary to what NSTIC requires, the linkability capabilities of the hub are being ignored. Likewise, the GOV.UK Verify specification allows the hub to learn information that can be used to relate activities of the same user. However, the Identity Assurance Principles ask that “No relationships between parties or records should be established without the consent of the Service User” and further related documentation makes explicit that “it is important to understand the impacts that would result should the service be compromised [CES12].”

Leaving the hub outside of the scope of privacy and security goals triggers serious problems, which we evidence in this paper. We reach the troublesome conclusion that the actual systems are in sharp opposition to privacy guidelines, including certain NSTIC requirements [NST13] and Identity Assurance Principles [Pri14], e.g., related with minimization of “relationship data.” We find major flaws that make the systems vulnerable to numerous privacy and security attacks, leading to the conclusion that the FCCX and GOV.UK Verify solutions, as currently inferred, may actually *degrade* the privacy of citizens. Specifically, the excessive trust placed on the hub could be notably used to support undetected mass surveillance.

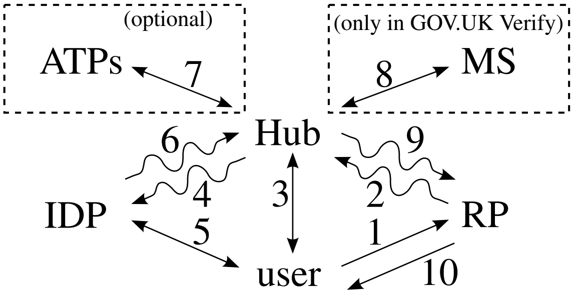

**Fig. 1. Flow of an identification transaction in FCCX and GOV.UK Verify**. Legend: ↗ ↙ (transmission); ↕ ⤡ ⤢ (interaction); ⤣ ⤦ ⤥ ⤤ (redirection through the user web-client).

On a more positive note, we show that, even within the assumed structural constraints, more resilient hub-based brokering alternatives are possible. We propose solutions that are deployable and resolve the privacy and security issues that we identify in the current FCCX and GOV.UK Verify. Furthermore, our solutions can balance privacy and security with external envisioned requirements of auditability and the ability to perform selective forensic inspections.

**Contributions.** Our paper offers several main contributions:

- We distill the essential aspects of an identification transaction use-case. This allows us to reason about the privacy and security properties offered and broken by FCCX and GOV.UK Verify.
- We highlight the exacerbated ability of the FCCX and GOV.UK Verify hubs to link events of the same user across different RPs, in spite of these systems advertising privacy through (a restricted notion of) unlinkability. We show how to achieve (a stronger notion of) unlinkability, while allowing *auditability* and *selective forensic disclosure*.
- We describe a basic solution to avoid visibility of *private identifiable information* by the hub. Surprisingly, this is not part of the initial development of FCCX or GOV.UK Verify.
- We discuss the lack of resilience, in FCCX and GOV.UK Verify, against a temporarily compromised hub being able to impersonate users accessing RPs, and propose solutions.

**Roadmap.** The paper proceeds as follows. Section 2 defines the brokered identification problem, the intervening parties and the type of intended identification transactions. Section

3 infers aims and features of FCCX and GOV.UK Verify and enumerates additional desirable system properties. Section 4 describes how FCCX and GOV.UK Verify implement an identification transaction. Section 5 diagnoses serious privacy and security problems and suggests respective solutions. Section 6 discusses concerns about overreach to private information and concludes with recommendations.

# 2 Background

We describe the entities involved in hub-based *brokered identification* (§2.1), the role of *pseudonyms* and *attributes* in the envisioned *identification transaction* use-case (§2.2), and the approach followed by FCCX and GOV.UK Verify (§2.3).

## 2.1 The identity ecosystem

The problem of credential management arises in the context of an *identity ecosystem* composed of entities with different roles. Below, we borrow some wording from NSTIC and the FCCX documents [The11, Uni13].

- A *user*, also known as individual, citizen, or customer, is a "person engaged in online transactions;" the term *subject* can also be used to include non-person entities.
- A *Relying Party* (RP) "makes transaction decisions based upon ... acceptance of a subject's authenticated credentials and attributes." The term can be used to denote "individualized federal agency systems and applications," e.g., online government tax services, but its use can also be extended to private-sector service providers.
- An *Identity Provider* (IDP) is "responsible for establishing, maintaining and securing the digital identity associated with" a subject. It can be a "non-federal credential provider" approved by an accreditation authority to provide *authentication assertions* up to a certain *level of assurance* (LOA). Increasing LOA values (1, 2, 3, 4) increase the "level of confidence that the applicant's claimed identity is their real identity" – requisites vary with the country (e.g., US [FF11] and UK [CES14]).
- An *Attribute Provider* (ATP) is "responsible for ... establishing ... identity attributes," such as"legal name, current address, date of birth, social security number, email address."

## 2.2 Identification transactions

We consider identification transactions where a relying party (RP) identifies and authenticates a user based on the ability of the user to authenticate to an identity provider (IDP). For the RP, the goal of identification is to learn: (i) a *persistent anonymous identifier* (notation found in [Cyb11, Joh12, USP14]) of the user, hereafter simply denoted as *user pseudonym*, which is always the same when the user authenticates through the same account at the IDP; and/or (ii) some personal attributes of the user, validated by the IDP, e.g., name, birth date, address. For the RP, the goal of authentication is to (i) gain confidence that the learned values are valid for the user with whom a session is established; and (ii) receive a *certified* assertion to that effect.

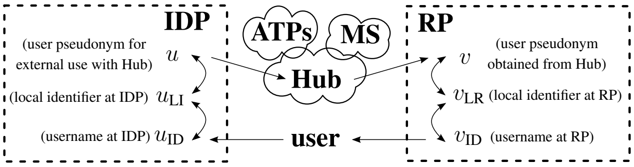
IDP
ATPs
MS
RP
(user pseudonym for external use with Hub) $u$
Hub
$v$ (user pseudonym obtained from Hub)
(local identifier at IDP) $u_{\text{LI}}$
$v_{\text{LR}}$ (local identifier at RP)
(username at IDP) $u_{\text{ID}}$
user
$v_{\text{ID}}$ (username at RP)

**Fig. 2. Relation between user identifiers at IDP and RP.**

**Link to a user account at the RP.** A user may want to create or reconnect into a personal account at some RP. However, the user only knows her own username (e.g., an email address) at an IDP, and how to authenticate to the IDP (e.g., using a password). Internally, the IDP is able to associate that username with other identifiers of the same user. In particular, for each brokered identification scheme the IDP derives a new user pseudonym for external use with the respective hub. Since it seems that in practice a single hub is being developed for each of FCCX and GOV.UK Verify, hereafter we use the symbol $u$ without ambiguity to denote the user pseudonym defined by the IDP for interaction with the hub. In both systems, $u$ is supposed to be pseudo-random and remain the same for all transactions with the same user.

Then, the RP learns from the hub a user pseudonym $v$ persistently associated with the pair ($u$, RP), but not revealing anything about $u$. The RP can then internally associate the received pseudonym to a local user account, which may contain further user information. We will later describe how the pseudonyms are transformed as part of the brokered identification scheme. The type of transformation and the (in)visibility of these pseudonyms is essential in determining the privacy of the scheme, namely the possible types of (un)linkability that can be inferred from user pseudonyms.

**Attribute integration.** As part of *identity proofing*, it may be necessary to transmit and/or verify attributes within a transaction, e.g., confirm minimal age before letting a user create an account. In the simplest case, the initial IDP (with whom the user authenticates) is able to validate the necessary attributes. In more complex interactions, attribute integration might have to involve *attribute enrichment*, i.e., attributes from different attribute providers (ATPs). For instance, a user logging into a hospital system would use the IDP to prove their identity, but could need an ATP to show proof of insurance. For simplicity, we mostly ignore the case of additional ATPs – this is not yet fully defined or developed in FCCX or GOV.UK Verify ([Gen14, Q&A],[Ide13a, Step 9]).

## 2.3 Brokered identification

**Hub and Matching Services.** FCCX and GOV.UK Verify propose using a brokered identification scheme, where an online central hub actively mediates the communication and ensures interoperability between RPs and IDPs. The systems use standardized web back-end technologies, such as the SAML assertion language [OAS05] and the XML Encryption [IDS02] and Signature [BBF+13] standards. Upon receiving an authentication request from the RP (link 2 in Fig. 1), the hub helps the user choose an IDP (link 3) and then redirects the user (and the request) to the chosen IDP (link 4). Then, as a result of the user authenticating to the IDP (link 5), the IDP sends to the hub a signed assertion conveying a user pseudonym and attributes (link 6). In both systems, the hub (and in GOV.UK Verify also the MS) sees the pseudonym and the attributes in the clear. However, the systems differ in how they transform the user pseudonym between the IDP and the RP, and how they recertify the authentication assertion:

- In FCCX, the hub transforms the user pseudonym $u$ received from IDP into a new user pseudonym $v$ for the RP, varying with RP. The hub removes the signature of the IDP (and other metadata), re-signs the assertion, and sends it to the RP (link 9).
- In GOV.UK Verify, the hub relays the assertion from the IDP to the *matching service* (MS) indicated by the RP (link 8). The MS validates the signature of the IDP and derives a new (locally generated) user pseudonym $v$ that is equal for all RPs that choose this MS. The MS also verifies that the locally-generated user pseudonym and attributes match to a local user account. Finally, the MS re-signs a new assertion and sends it to the hub (still (link 8)), who then re-signs the assertion and sends it to the RP (link 9).

**User limitation.** A main design constraint in FCCX and GOV.UK Verify is that the role of the user-agent (a web browser) in the protocol is substantially passive. The active participation of the user is limited to requesting a resource from the RP, selecting an IDP (from a list) and authenticating to the IDP. Communications between the RP and the hub, and between the IDP and the hub, are passively redirected through the user, e.g., using Web browser SSO HTTP POST [OAS05]. The relayed authentication requests and assertions use a "SAML 2.0 Web Browser SSO Profile" and are signed by the originator and encrypted for the intended recipient. Channel security, for communications with the RP, hub and IDP, is based on SSL/TLS [Int08]. Overall, these mechanisms prevent network observers and the user from viewing the exchanged user pseudonyms and attributes, and/or otherwise trivial message manipulation attacks.

**Out-of-scope alternatives.** Privacy aside, linking to a user account at the RP could be achieved by a direct connection between RPs and IDPs (i.e., intermediated by a passive user), as accomplished with *OpenID Connect* [Ope]. However, this would not hide the IDP and RP from one another, which is a main explicit goal in the FCCX and GOV.UK Verify context. Alternatively, using group signatures and anonymous credentials [ISO13] would allow IDPs to sign assertions that RPs could validate as signed by an entity from within a specified group, but without knowing who. However, on its own, this approach would not provide a privacy-preserving way to transform user pseudonyms between the IDP and the RP and, without an external broker, would require more user involvement to mediate the communication between the IDP and RP. A group-signature based approach would also require group membership management and/or a mechanism for detection and isolation of compromised IDPs that could otherwise taint the trust in the system.

More interesting solutions would be possible if the user could actively aid the brokering between IDP and RP, e.g., using cryptographic protocols based on privacy-enhancing technologies. It could take advantage of a trusted setup, e.g., tamper-resistant trusted hardware and/or open-source software authenticated by a party trusted by the user (as already happens when choosing a web browser). This approach may provide a promising alternative to the designs imposed by FCCX and GOV.UK Verify. However, our goal in this paper is to present the actual FCCX and GOV.UK Verify mechanisms and then show that their privacy and security problems can be repaired within their own design constraints.

# 3 System properties

The available FCCX and GOV.UK Verify documentation is incomplete in several aspects. The FCCX solicitation partially describes certain desirable privacy and security properties, but does not specify the transaction protocol. The GOV.UK Verify specification defines protocol steps in more detail, but we also do not find a well-defined list of desired properties. Therefore, we need to infer, from their public descriptions, a set of properties that they seemingly intend to achieve (§3.1).

The privacy and security of FCCX and GOV.UK Verify rely on a fully honest and uncompromisable hub. In contrast, we argue that a good solution should be resilient even when the hub is *curious* (about what it sees) and/or *malicious* (about the actions it takes). In other words, we consider two types of corrupted parties: *honest-but-curious* (i.e., acting honestly during transactions, but curious to derive information from the observed communications) and *malicious* (capable of deviating from the protocol specification). This gives rise to a number of additional desirable properties beyond those inferred from the two analyzed systems (§3.2).

A good protocol should also be resilient against malicious collusion, e.g., between RPs, or between RP(s) and the hub. However, to satisfy forensic requirements, certain special cases of collusions (e.g., hub+IDP, or hub+IDP+RP) may be legitimately allowed to reverse certain privacy properties, e.g., unlinkability, under very well-defined circumstances such as a specific court order targeting a given individual. Regrettably, in both FCCX and GOV.UK Verify, the forensic capabilities of the hub as currently described could be abused and enable undetected mass surveillance.

Table 1 summarizes shortcomings of FCCX and GOV.UK Verify in fulfilling a number of these properties. The approaches proposed in this paper show that a different outcome can be achieved without dramatic design changes. We consider several aspects of unlinkability, as well as privacy of attributes, authenticity (i.e., resilience to impersonation), one-to-one traceability and selective forensic disclosure.

We discuss "unlinkability" associated with user pseudonyms, ignoring linkability related to side-channel information, such as timestamps and IP addresses. Their mitigation can be addressed in a lower-level specification and/or by prudent implementation guidelines and/or techniques outside of the scope of the system (see Appendix B).

| | | A | B | C | D | |
|---|---|---|---|---|---|---|
| **Properties** | **Systems** | **Direct connect** | **FCCX** | **GOV.UK Verify** | **This paper** | |
| **Edge unlinkability within a transaction:** | | | | | | |
| RP identity is hidden from IDP | | NO | YES | | YES | 1 |
| IDP identity is hidden from RP | | NO | YES | | YES | 2 |
| **Unlinkability across same-user transactions:** | | | | | | |
| **Weak** | hub/MS cannot link user across RPs | — | NO | | **YES** (§5.1.1) | 3 |
| | hub/MS cannot link user across IDPs | — | Y/N* | NO | **YES** (§5.1.3) | 4 |
| **Strong** | hub cannot link user across transactions | — | NO | | **YES** (§5.1.2) | 5 |
| **Edge** | Change of RP is hidden from IDP | NO | YES | | YES | 6 |
| | Change of IDP is hidden from RP | NO | N/Y* | NO | **YES** (§5.1.3) | 7 |
| | Colluding RPs cannot link pseudonyms | YES$^\dagger$ | YES | NO$^\ddagger$ | YES (§5.1.4) | 8 |
| **Other properties:** | | | | | | |
| Atttribute privacy from hub | | — | NO$^\#$ | | **YES** (§5.2) | 9 |
| Authenticity if malicious hub | | — | NO | | **YES** (§5.3) | 10 |
| One-to-one traceability | | — | NO | | **YES** (§5.4) | 11 |
| Selective forensic disclosure$^\|$ | | — | NO | | **YES** (§5.4) | 12 |

**Table 1. Properties across types of solutions.** Legend: "**YES**" is good for privacy; the two alternatives in cell B4 (Y=Yes or N*=No) are entangled with the complementary alternatives in cell B7 (N or Y*) – the second alternative (*) results from an optional *account linking* functionality [Uni13]; cell A8 (†) assumes that the IDP sends different user pseudonyms to different RPs; cell C8 (‡) considers RPs that have chosen the same MS; in cell BC9 (#), privacy is broken even if the hub is honest-but-curious; in line 12 (‖), the property is meant in opposition to total disclosure by default.

## 3.1 Inferred properties

**Authenticity.** Upon completion of a transaction, the relying party (RP) is assured that it has established a session with the user from whom it holds a fresh claim, and that the claims are valid (namely the user pseudonym $v$ and attributes). A "session" can mean, for instance, a TLS/SSL encrypted tunnel. The root of trust for authenticity rests with the identify providers (IDPs) and with the hub or matching service (MS). Specifically, the hub/MSs trusts the authentication assertions received from IDPs and ATPs; the RP trusts the authentication assertions received by the hub/MS. In GOV.UK Verify the RP chooses which MS to trust, whereas in FCCX there is a single hub in which to rely. However, we will show that in both systems the authenticity can be broken by a malicious hub.

**Edge unlinkability within a transaction.** A key idea behind (privacy-preserving) brokered identification is to shield the "edges" of the authentication system (i.e., IDPs and RPs) from knowing about each other. We say that the system has *edge unlinkability within a transaction* if: (i) the IDP does not learn about who is the RP and MS; (ii) the RP does not learn about which IDP authenticated the user; and (iii, iv) the IDP and RP do not learn about the user pseudonyms at the other party.

In spite of edge unlinkability, in FCCX and GOV.UK Verify the hub knows who is the IDP and RP in each transaction; it is unclear to us if in GOV.UK Verify the MS knows who the RP is, but it knows the IDP.

**Traceability.** If an auditor challenges the legitimacy of an action taken by a party, with respect to a transaction, then the party should be able to justify it based on a verifiable preceding action. Specifically, for each authentication request sent from the hub to the IDP, the hub must have a respective request signed by the RP; for each authentication assertion sent from the IDP to the hub, the IDP must have a respective request signed by the hub; for each assertion sent from the hub to the RP, the hub must have a respective assertion signed by the IDP; and for each user login at an RP, the RP must have a respective assertion signed by the hub.

While not explicitly discussed in the design documents, we infer the intention of *traceability* from the use of signatures in the proposed FCCX and GOV.UK Verify systems. Traceability is useful toward allowing *auditability* of the behavior of each isolated party. However, it is possible to achieve traceability more meaningfully than in FCCX and GOV.UK Verify, namely to promote better *accountability*. Specifically, we suggest solutions that achieve *one-to-one* traceability; e.g., if the hub justifies one authentication request or assertion on the basis of another authentication request or assertion, then such justification is the only one possible.

## 3.2 Additional desirable properties

**Unlinkability by the hub.** The hub should not be able to link the same user across different transactions. Since the hub is part of the brokered identification system, this property is required to satisfy the notion intended by NIST: "*unlinkability assures that two or more related events in an information-processing system cannot be related to each other*" [NIS13]. Also, NSTIC requirements include that "organizations shall minimize data aggregation and linkages across transactions" [NST13, Req. 5], and IDAP principles state that "no relationships between parties or records should be established without the consent of the Service User." [Pri14, Principle 7.5]. We consider two weaker nuances and a strong notion:

- *Weak unlinkability across RPs:* the hub cannot link transactions of the same user (as defined by an account at an IDP) across different RPs. Since a user account at the IDP can be used to access many RPs, the user pseudonym defined by the IDP can be considered a global persistent identifier and thus should not be learned by the hub. Neither FCCX nor GOV.UK Verify satisfy this. In the UK, allowing global persistent identifiers conflicts with the political sensitivities that arguably lead to the rejection of identity cards [BD11]. In the US, NSTIC specifically calls for "privacy-enhancing technology that ... *minimizes the ability to link credential use among multiple RPs*" [NST13, Req. 5].
- *Weak unlinkability across IDPs:* the hub or MS cannot link different transactions facilitated by different user accounts at one or more IDPs leading to the same user account at a given RP. In GOV.UK Verify such linkage is performed by default by the MS (chosen by the RP), based on the user attributes. The user thus does not control who has the ability to link, nor when to allow linking—a clear privacy deficiency. FCCX offers, via an optional *account linking* feature, a tradeoff: endow the hub with the capability of *linkability across IDPs*, in exchange for allowing authentication to each RP from different accounts at IDPs. We will show that this tradeoff can be avoided.
- *Strong unlinkability:* the hub cannot link transactions where the same user account at an IDP is being used to access the same user account at an RP. Neither FCCX nor GOV.UK Verify satisfy this property.

**Edge unlinkability across transactions.** We earlier discussed edge unlinkability within a transaction; the notion can be extended *across* transactions as follows:

- *Across two transactions with the same user account at an IDP:* the IDP does not learn whether the accessed RP has changed or not. This property can be inferred from the FCCX and GOV.UK Verify designs.
- *Across two transactions with the same user account at a RP:* the RP does not learn whether the assisting IDP has changed or not. This property is achieved in FCCX when using an "account linking" option, but with a privacy tradeoff (which we will show how to avoid);
- *Across transactions with the same user account at an IDP but different RPs:* (i) several colluding RPs cannot link the same user based on their lists of user pseudonyms; (ii) if several RPs colluding together know, from an external source, that respective user pseudonyms correspond to the same user, they are still not able to predict anything about the user pseudonym at another RP. This is satisfied in FCCX, but not in GOV.UK Verify where different RPs (that have chosen the same MS) receive the same user pseudonym.

**Attribute privacy.** The visibility of personal identifiable information, namely attributes, should be reduced to the bare minimum necessary for the purpose of each party and as consented by the respective user. For example, relying parties should learn nothing more than necessary and requested (e.g., an age predicate, instead of a date of birth). We are conveying that there should exist capability to deal with predicates of attributes, but the actual definition of what is "necessary" is outside the scope of our system model. As for the hub, since its role is to help the interoperability of transactions, supposedly without interest about user information, it should not have visibility into the attributes being exchanged or verified. This is required by the FCCX procurement, but is not currently achieved [Gen14, Q&A 2.3]. The GOV.UK Verify specification simply defines a protocol where the hub and MS have visibility of attributes. In GOV.UK Verify the MS explicitly uses some attributes to help link the user into a local account.

**Resilience against impersonation.** In spite of the trust placed in the hub to broker a transaction, a maliciously compromised hub should not be able to break authenticity. It should not be able to gain access to a (honest) user account at a (honest) RP. However, we will show that in FCCX and GOV.UK Verify a compromised hub is able to impersonate users.

**Selective forensic disclosure.** NSTIC [NST13, Req. 22] and IDAP [Pri14, Princ. 9] contain provisions about forensic capabilities and exceptional circumstances. They consider "forensic capabilities to ... permit attribution" and possible "exemption from ...[other] Principles that [the exemption] relates to the processing of personal data" if "necessary and justifiable in terms of" foreseen exceptions to the "Right to respect for private and family life" [Eur10]. With this in mind, a desirable property might be to have the ability to do a limited reversal of weak or strong unlinkability (or attribute privacy) in special cases where a subset of entities are compelled to aid in an adequate investigation, e.g., upon being served a subpoena.

Assume the hub pinpoints a transaction related to a certain triplet (IDP, user, $\text{RP}_1$), where we mean "user" as defined by $u$. We envisage two types of selective forensic disclosure:

- **Coarse-grained.** Compelled collaboration of the IDP may allow the hub to gain full linkability of past (logged) transactions of the selected user with any RP (i.e., pinpoint all such transactions), but without affecting the unlinkability of other users.
- **Fine-grained.** Compelled collaboration of the IDP and some $\text{RP}_2$ with the hub may allow the hub to pinpoint past (logged) transactions of the same user with this $\text{RP}_2$, but (i) without IDP learning who is $\text{RP}_1$ and $\text{RP}_2$, (ii) without the hub learning about any other transactions of the user with any RP other than $\text{RP}_2$, and (iii) without breaking unlinkability of other users. In other words, edge unlinkability is preserved and weak unlinkability is selectively broken to investigate a user in connection to one or several selected RPs, without leakage about interactions with other RPs. If $\text{RP}_1$ is $\text{RP}_2$, then the IDP does not need to collaborate.

In sharp contrast, in FCCX and GOV.UK Verify both types of leakage happen by default and without any need for collaboration. This is a serious vulnerability, and could open the way to undetected mass surveillance.

# 4 Inferred protocols

The high-level protocol flow of a transaction in FCCX and GOV.UK Verify was depicted in Fig. 1. In Fig. 3 we describe the respective steps in more detail.

To simplify, we leave implicit some elements of metadata and omit verifications that must be done at each party.

1. **Start.** The user requests a resource from the RP (1).
2. **Initial authentication request.** The RP selects a SAML identification number (ID) $n$ (2), and uses SAML syntax to build an authentication ("authN" in the figure) request, also containing (not shown) the required level of assurance and other metadata. In GOV.UK Verify the RP also specifies the MS (3). The RP signs the request and sends it encrypted to the hub, via user redirection (4).
3. **Select IDP.** The hub asks the user to select an IDP (5).
4. **Relay authentication request.** The hub prepares a new authentication request, removing metadata that could identify the RP. In FCCX it is not clear if the request ID ($n'$) of the new request is equal to the one ($n$) received from the RP (6) (e.g., to prevent it from being used as a covert-channel from RP to IDP). In GOV.UK Verify, this ID number does not change (7) – it will be visible by all parties (except the user) across the transaction. The hub signs the new request and sends it encrypted to the IDP, via user redirection (8).
5. **Authenticate user at IDP.** The IDP and user perform an arbitrary (possibly multi-round) authentication protocol (9).
6. **Initial authentication assertion.** The IDP determines a pseudo-random user pseudonym $u$, persistently associated with the local user account (and none other), and defined specifically for brokered transactions with this hub (10). Next, the IDP builds an authentication assertion that includes the authentication request ID ($n$'), the user pseudonym ($u$), some attribute values ($atts$) and some contextual information ($ctx$: the level of assurance, authentication type and other transient attributes such as the IP address of the *user*) (11). We comment in Appendix A about the set of default attributes in FCCX and GOV.UK Verify. The IDP signs and encrypts the assertion and sends it to the hub via user redirection (12). The hub can view all the data in the assertion, including the user pseudonym ($u$) defined by the IDP and the attributes.
7. **Attribute enrichment.** Depending on the authentication request from RP (and assuming user consent), the transaction may involve integration of attributes obtained from several ATPs (13). We consider here a case where such integration is not needed and defer further comments to Appendix A.
8. **Matching to local account (only in GOV.UK Verify).**

**8a.** The hub signs the assertion $a_0$ received from the IDP, and sends encrypted to the MS (chosen by RP) the assertion and the two signatures (by the hub and by the IDP) (14). This is sent directly, using SAML SOAP binding, rather than via user redirection.

**8b.** The MS then "locally generates" a user pseudonym ($v$), as the SHA256-hash of the concatenation of the IDP identifier, the MS identifier and $u$ (15). The MS then uses $v$ to try to find a match to a local account identifier ($v_{\text{Lm}}$) unique to the user. If a match is not found, then the MS attempts to find a match based on the provided *matching data set* attributes (a

1. user → RP : request resource (1)

2. RP : $n \leftarrow$ SAML ID (2)

RP : $c = \{n\}$ (authN request) (3)

RP $\rightsquigarrow$ hub : $E_{\text{hub}}\,(c, \sigma_{\text{RP}}(c))$ (4)

3. hub ↔ user : Select IDP (5)

4. FCCX hub : $n' \leftarrow^{?} n$; $c' = \{n'\}$ (authN request) (6)

GOV.UK hub : $n' = n$; $c' = \{n'\}$ (authN request) (7)

hub $\rightsquigarrow$ IDP : $E_{\text{IDP}}(c', \sigma_{\text{hub}}(c'))$ (8)

5. IDP ↔ user$(u_{\text{ID}})$ : arbitrary authN protocol (9)

6. IDP : $u \leftarrow \text{Get}_{\text{Pseudonym}}(u_{\text{Li}}, \text{hub})$ (10)

IDP : $a_0 = \{n', u, atts, ctx\}$ (authN assertion) (11)

IDP $\rightsquigarrow$ hub : $E_{\text{hub}}\,(a_0, \sigma_{\text{IDP}}(a_0))$ (12)

7. hub ↔ ATP$s$ : (optional) attribute enrichment (13)

8. Only in GOV.UK Verify:

$8a$. GOV.UK hub → MS : $E_{\text{MS}}(a_0, \sigma_{\text{IDP}}(a_0), \sigma_{\text{hub}}(a_0))$ (14)

$8b$. MS : $v = \text{Hash}(\text{IDP}, \text{MS}, u)$ (15)

MS : $v_{\text{Lm}} \leftarrow \text{Find}_{\text{match}}(v, atts)$ (16)

MS : Store$((v_{\text{Lm}}, v))$ (17)

MS : $a_1 = \{n, v, atts, ctx\}$ (authN assertion) (18)

MS → GOV.UK hub : $E_{\text{hub}}(a_1, \sigma_{\text{MS}}(a_1))$ (19)

9. FCCX hub : $v \leftarrow \text{Get}_{\text{Pseudonym}}(u, \text{RP})$ (20)

FCCX hub : $a_1 = \{n, v, atts, ctx\}$ (authN assertion) (21)

FCCX hub $\rightsquigarrow$ RP : $E_{\text{RP}}\,(a_1, \sigma_{\text{hub}}(a_1))$ (22)

9. GOV.UK hub $\rightsquigarrow$ RP : $E_{\text{RP}}(a_1, \sigma_{\text{hub}}(a_1), \sigma_{\text{MS}}(a_1))$ (23)

10. RP : $v_{\text{Lr}} \leftarrow \text{Find}_{\text{match}}(a_1)$ (24)

RP → user : response (25)

**Fig. 3. Inferred protocols.** Legend: $x \rightarrow y$ (message sent from $x$ to $y$); $x \rightsquigarrow y$ (message sent from $x$ to $y$ via a user redirection); $x \leftrightarrow y$ (interaction between $x$ and $y$); $x \leftarrow^{?} y$ ($x$ is obtained from $y$, after some (unspecified) transformation); $\{...\}$ message with SAML-syntax (leaving implicit certain meta-data); $\sigma_x(z)$ (signature of content $z$ by entity $x$); $E_x(z)$ (encryption of content $z$, using the public key of $x$).

default set of attributes defined by GOV.UK Verify) (16). If a match is still not found (and if one was not required), then the MS may create a "temporary" local identifier ($v_{\text{Lm}}$). The MS then updates the mapping that it maintains between identifiers, storing if necessary the pair ($v_{\text{Lm}}$,$v$) (17), and builds an assertion including the authentication-request ID $n$, the locally generated $v$ (associated with the user account at IDP), the attributes $atts$ and the context $ctx$ (18). The MS signs the assertion and sends it encrypted to the hub (19).[1]

**9. Relay authentication assertion** In FCCX, the hub directly converts the pseudonym received from the IDP into a pseudonym for the RP (different and persistent for each RP) ($v$) (20). The hub then uses $v$, instead of $u$, to build a new assertion $a_1$ for the RP (21). The hub re-signs the assertion, encrypts it and sends it to the RP via user redirection (22).
In GOV.UK Verify, the hub can view all the data in the assertion received from MS, including the new user pseudonym. The hub re-signs the assertion, encrypts it and sends it to RP, along with the signature from the MS (23).

**10. Final response** The RP processes the assertion received from the hub, possibly finding a local identifier for the user account (24). Finally, the RP responds to the user, possibly granting or denying access to the requested resource (25).

# 5 Problems and solutions

We show that in FCCX and GOV.UK Verify a corrupt *hub* may violate with impunity many of the properties defined in §3.

## 5.1 Linkability of same-user transactions

### 5.1.1 Weak unlinkability across RPs

In FCCX and GOV.UK Verify, the hub has full visibility of the user pseudonym ($u$) defined by the IDP. Thus, the hub—and whoever can gain control over it, legitimately or not—can link transactions of the same user, as defined by a user account at an IDP, across different RPs. This gives the hub excessive visibility into the activities of all users. It also violates the selective forensic disclosure property, by allowing linkability without the help of the respective IDPs or RPs. Even if there are provisions about not storing some data (e.g., in GOV.UK Verify), a system should be at least secure in a honest-but-curious adversarial model, where seeing is equivalent to storing. In FCCX, the hub is supposed to log activities related to all transaction steps for at least one year online and 7.5 years offline [Uni13]. In any case, we advocate that clear information should be made public about existing security-related logging.

We propose a solution for weak unlinkability (by the hub) across RPs, involving interaction between IDP and the hub. The solution is directly applicable to FCCX, but not to GOV.UK Verify where the user pseudonym is transformed by the MS. Anyway, we propose that the GOV.UK Verify structure be changed to integrate our solution, since we also show

[1] The MS is sending to the hub a pseudonym $v$ that does not vary with the RP; we ponder if instead it might have been intended that the MS would send a pseudonym derived from $v_{\text{Lm}}$, independently of IDP. As described, the local matching performed by MS (16-17) has no obvious advantage to RP. We will show that a better alternative is possible (§5.1.3). The MS also appears to store user attributes (not shown in the figure), since it needs them when matching is not possible based only on the pseudonym $u$.

(§5.3) that the MS does not ensure authenticity against a malicious hub, and poses additional threats against unlinkability.

#### 5.1.1.1 Initial intuition

Linkability by the hub of user pseudonyms across RPs can be avoided by hiding $u$ from the hub, and with the hub learning $v$ (the user pseudonym to send to RP) as a pseudorandom value. The pseudonym $v$ is persistent for each pair $(u, r)$, where $r$ is defined by the hub as an identifier of the RP for interactions with the IDP. Also, the IDP must not learn anything about $r$ and $v$. This is an instance of a secure-two-party computation (S2PC), where the hub learns a function of combined inputs but neither hub nor IDP learn the input or output of the other. The IDP provides $u$ as input; the hub provides $r$ as input and receives $v$ as output. The function can be a block cipher, with the key being the user pseudonym $u$ held by the IDP, and with the plaintext being the RP identifier $r$ held by the hub. By definition, the output $v = \text{Cipher}_u(r)$ of a secure block cipher is indistinguishable from random, if the key $(u)$ is random. The RP identifier $r$ must be unpredictable by the IDP so that the output $v$ is also unpredictable by the IDP.

For example, (Yao-protocol-based) S2PC of the AES block-cipher is a *de facto* standard benchmark in S2PC research [DLT14]. The respective circuit requires 6,800 garbled gates [Bri15], which is about 13 times less than for a SHA256 circuit. Recent techniques allow a significant amortization of the computational complexity of S2PC in a multiple execution setting, i.e., when evaluating many times the same circuit, and allowing various tradeoffs between offline and online phases [HKK$^+$14, LR14]; this applies here as the hub and each IDP are involved in many transactions. Recent work also shows how to minimize the number of communication rounds [AMPR14]. Other block-ciphers may have circuit designs that enable even more efficient S2PC implementations [ARS$^+$15].

While a S2PC-AES execution is slower than a simple exchange of signatures, available benchmarks show that it can be achieved under a second [FJN14]. Furthermore, after the user authenticates to the IDP, the IDP and the hub can execute the S2PC "behind the scenes," without user intervention. In other words, the added latency is unlikely to add a considerable usability burden. Even if S2PC-AES may not be a perfect solution to achieve weak unlinkability across RPs, it is a proof-of-possibility demonstrating that weak unlinkability is achievable within the imposed constraints and thus ought to be required in privacy-preserving brokered identification systems.

**Adding traceability.** To achieve traceability, the IDP should also sign an assertion that allows the hub to justify subsequent actions related to $v$ in this transaction. While theoretically this could be achieved by embedding a signature calculation within the generic S2PC module, it would prohibitively increase the cost of the computation. To avoid this, we propose a way in which signatures can be computed outside of the S2PC module. Each party signs and receives a signature of commitments that hide and bind the parties to the respective inputs used and the outputs obtained in the S2PC. For example, given the secrecy constraints, the IDP does not sign the unknown pseudonym $v$, but rather a commitment efficiently obtained in the S2PC. The commitment hides $v$ from the IDP, but binds the hub to the value, preventing association with any other value. The process is also applied to the inputs of both parties, to commit the hub to a RP identifier ($r$), and to commit the IDP to the user pseudonym ($u$). These commitments need not be *opened*, except in an eventual audit that so requires.

The needed commitments can be obtained by a specialized S2PC protocol that directly provides commitments of the inputs and outputs of both parties (e.g., [Bra13]). However, in the interest of generality, below we also propose an alternative applicable to any generic S2PC protocol of Boolean circuits. The devised mechanism combines (any type of) cryptographic commitments, computed prior to the S2PC module, with the S2PC of a circuit augmented with efficient randomize-then-mask operations (⊗-then-⊕). In this approach, a party may still use in the S2PC inputs different from those committed, but traceability is not affected because an audit would detect an inconsistency with overwhelming probability.

Below, we sketch a protocol with three stages: (i) produce cryptographic commitments ($\mathcal{C}$), prior to the S2PC; (ii) execute S2PC of a block-cipher circuit augmented with randomize-then-mask operations; (iii) sign the obtained masked values, after the S2PC. We then give a step-by-step description, as depicted in Fig. 4, for simplicity omitting some metadata necessary in a concurrent setting, and some verifications (e.g, message syntax, expected values and signatures).

#### 5.1.1.2 Protocol sketch and further intuition

1. **Commitments.** Each party cryptographically commits to her inputs and to several random *masks* (or "blinding factors"), i.e. bit-strings created to obfuscate other bit-strings.
2. **S2PC.** The hub and IDP engage in a S2PC, where each party learns masked versions ("*maskings*") of the input and output of both parties. The maskings are defined so that their result is not predictable before the S2PC. We achieve this with a ⊗-randomize then ⊕-mask operation, with contributions (randomizer and mask) from both parties, for suitable ⊗ and ⊕ binary operations. The ⊗-then-⊕ somewhat resembles a one-time *message authentication code*. Essentially, any inconsistency between the values used in

the S2PC and the values previously committed is detectable in an audit action that requires opening the commitments.

3. **Signatures.** After the S2PC, the hub sends to IDP a signature of all maskings from both the hub and IDP. The IDP then reciprocates with a signature of the same elements, and by opening one of the two masks that was blinding the final output $v$ of the hub (who knows the other mask).

**Input masking.** As an example, let $u$ be the original private input of the IDP. We augment the protocol so that the IDP can later prove to an auditor that $u$ was indeed an input of the S2PC. As a first step, prior to the S2PC, the IDP produces and sends a commitment $\chi_u \leftarrow^{\$} \mathcal{C}(u, \gamma_u)$ to the hub, where $\gamma_u$ is a secret $\oplus$-mask. The commitment binds the IDP to the values $u$ and $\gamma_u$, while hiding them from the hub. Then, let the IDP maliciously use values $u^*$ and $\gamma_u^*$ as input in the S2PC, where $(u^*, \gamma_u^*) \neq (u, \gamma_u)$ (i.e., one or both elements are different). Then, let the S2PC (besides the original computation that it was supposed to do) also compute and reveal the value $\delta_u = (u^* \otimes \beta_u) \oplus \gamma_u^*$, where $\beta_u$, used as auxiliary input by the hub, is a $\otimes$-randomizer also revealed as an auxiliary output of the S2PC. In other words: the party that knows an input value to be committed in the S2PC selects an $\oplus$-mask, whereas the other party selects an $\otimes$-randomizer; then, the S2PC reveals the overall masking ($\delta$) and the randomizer ($\beta$) to both parties, but neither the mask ($\gamma$) nor the committed value. As part of the overall protocol, both parties are then compelled to sign the values $\delta_u$, $\beta_u$ and $\chi_u$, i.e., only if obtained from a successful S2PC. The properties of $\oplus$ (e.g., bit-wise XOR) and $\otimes$ (e.g., multiplication in an adequate Galois field of characteristic 2) are chosen such that, for any $u$, $\gamma_u$ and $\delta_u$, if $\beta$ is sampled uniformly at random, the probability that $(u \otimes \beta) \oplus \gamma_u$ is equal to $\delta_u$ is negligible.[2] Thus, if the IDP is audited, it will with overwhelming probability be caught cheating.

Assume that $u$ has $\kappa$ bits (e.g., 128). While $\otimes$ could be multiplication in a field of order $2^\kappa$, it can be simplified to a randomized hash function $\mathcal{H}_\beta(x) = x \otimes \beta$, indexed by $\beta$, e.g., selecting a function from a *universal hash family* $\mathcal{H}$. It is enough to have the collision probability, which determines the binding property, be negligible in an acceptable statistical parameter $\sigma$, e.g., 40 bits. Thus, the hash may compress the input into, say, 40 bits. We leave for future work possible protocol adjustments that may reduce the burden imposed by concrete implementations of $\otimes$.

**Output masking.** The method is slightly different for committing an output, namely $v$ for the Hub, because it cannot be committed prior to the S2PC. Instead, the protocol outputs for both parties a double $\oplus$-masking of $v$, as well as commitments to each such mask. Each party (IDP and hub) knows one such mask, which is used as an augmented input of the S2PC. Later, once the IDP reveals one mask and proves it correct, the hub (who knows the other mask) learns $v$. Since both parties sign the masking and the commitments of the respective masks, the hub can prove to an auditor the value $v$, namely because it can also prove, by the method described for inputs, the correct value of the masks. This scheme ensures that the hub only learns $v$ after sending a respective signature to the IDP.

**Security properties.** Unlinkability follows directly from the hiding properties of the commitment schemes, of the $\oplus$-maskings and the block cipher. By the S2PC properties, the IDP learns nothing about $v$ or $r$. By the pseudo-randomness of a block-cipher with a random secret key, the hub learns nothing about $u$. Traceability for each party follows from the signature by the other party, containing values that the party can prove being unequivocally associated with only one particular input and/or output. Specifically, in a later audit phase, a party may open the $\oplus$-mask to an auditor, to allow verifiable unmasking of an input used and/or output received in the S2PC. Since the $\otimes$-randomizers lead to unpredictable values being masked, the commitment of values different from those used in the S2PC would lead to an inconsistent verification.

#### 5.1.1.3 Detailed procedure

**Initial inputs.**

The hub and the IDP agree on a session identifier (26). (See §C for a possible mechanism.) The private input of the IDP is a user pseudonym $u$, obtained after user authentication. The private input of the hub is an RP identifier $r$ (27).

**Randomizers, masks and commitments.** Each party (hub and IDP) selects random elements as follows (28-29): a $\oplus$-mask ($\alpha$ and $\alpha'$) for the final output $v$ of the hub; a $\otimes$-randomizer ($\beta$) for each original input ($u$, $r$) of the other party and for each $\oplus$-mask ($\alpha$ and $\alpha'$) of the final output of the hub; a $\oplus$-mask ($\gamma$) to apply to their own $\otimes$-randomized elements. For the purpose of this description, the $\oplus$-masks $\alpha$ and $\alpha'$ are hereafter also considered respective inputs of the hub and IDP. Then, for each input of each party ($r$ and $\alpha$ for the hub; $u$ and $\alpha'$ for the IDP), the party cryptographically commits to the value and to the respective $\oplus$-mask ($\gamma$). (30-31). Except for the commitment $\chi_{\alpha'}$ of the $\oplus$-mask used as input by the IDP, which will only temporarily $\oplus$-mask the output of the hub, the other commitments ($\chi_u$, $\chi_r$, $\chi_\alpha$) will not be opened within

[2] In particular, this implies that $\otimes$ and $\oplus$ are non-commutative, assuming that $\oplus$ is not collision-resistant in respect to the pair of inputs, i.e., assuming that it is feasible to find values such that $x \oplus \gamma = x^* \oplus \gamma^*$.

this protocol. They might (eventually) be opened later, to an auditor that requires knowing the values used in the S2PC.

**S2PC.** The parties then engage in a S2PC, having as respective inputs the original private inputs $(u, r)$ of the protocol and the selected random values (32). The S2PC internally computes:

- for each input $(u, r, \alpha, \alpha')$ of each party, the $\otimes$-randomization (using the randomizer known by the other party) followed by the $\oplus$-masking (using the mask known by the respective party) (33-36). The sequence of four maskings is denoted by $\Delta$ (37).
- the double $\oplus$-masking of the user pseudonym $(v)$ to be eventually learned by the hub (38).

$$\textbf{Common input: } n', \sigma_{\text{hub}}(\{n'\}) \text{ (session ID)} \tag{26}$$

$$\textbf{Initial private input} : \text{IDP}[u], \text{hub}[r] \tag{27}$$

**Procedure:**

$$\text{IDP} : \rho_I = (\gamma_u, \beta_r, \beta_\alpha, (\alpha', \gamma_{\alpha'})) \leftarrow^{\$} \{0,1\}^{3\sigma+\kappa+\sigma} \tag{28}$$

$$\text{hub} : \rho_H = (\beta_u, \gamma_r, (\alpha, \gamma_\alpha), \beta_{\alpha'}) \leftarrow^{\$} \{0,1\}^{2\sigma+\kappa+2\sigma} \tag{29}$$

(Only $\alpha$ and $\alpha'$ have length $\kappa$; the others have length $\sigma$)

$$\text{hub} \to \text{IDP} : \chi_r \leftarrow^{\$} \mathcal{C}(r, \gamma_r), \chi_\alpha \leftarrow^{\$} \mathcal{C}(\alpha, \gamma_\alpha) \tag{30}$$

$$\text{IDP} \to \text{hub} : \chi_u \leftarrow^{\$} \mathcal{C}(u, \gamma_u), \chi_{\alpha'} \leftarrow^{\$} \mathcal{C}(\alpha', \gamma_{\alpha'}) \tag{31}$$

$$\textbf{S2PC start: } \text{IDP}[u, \rho_I] \leftrightarrow \text{hub}[r, \rho_H] \tag{32}$$

$$\delta_u = (u \otimes \beta_u) \oplus \gamma_u \text{ ($\otimes$ed-$\oplus$ed input of IDP)} \tag{33}$$

$$\delta_r = (r \otimes \beta_r) \oplus \gamma_r \text{ ($\otimes$ed-$\oplus$ed input of hub)} \tag{34}$$

$$\delta_\alpha = (\alpha \otimes \beta_\alpha) \oplus \gamma_\alpha \text{ ($\otimes$ed-$\oplus$ed 1$^{\text{st}}$$\oplus$-mask for output of hub)} \tag{35}$$

$$\delta_{\alpha'} = (\alpha' \otimes \beta_{\alpha'}) \oplus \gamma_{\alpha'} \text{ ($\otimes$ed-$\oplus$ed 2$^{\text{nd}}$$\oplus$-mask for output of hub)} \tag{36}$$

$$\Delta \equiv (\delta_u, \delta_r, \delta_\alpha, \delta_{\alpha'}) \tag{37}$$

$$v' = \text{BlockCipher}_u(r) \oplus \alpha \oplus \alpha' \text{ ($\oplus$ed-$\oplus$ed output of hub)} \tag{38}$$

$$\textbf{S2PC end: } \text{IDP}[\Delta, \beta_u, \beta_{\alpha'}, v'], \text{hub}[\Delta, \beta_r, \beta_\alpha, v'] \tag{39}$$

$$\text{IDP}, \text{hub} : X \equiv (\chi_u, \chi_r, \chi_\alpha, \chi_{\alpha'}), B \equiv (\beta_u, \beta_r, \beta_\alpha, \beta_{\alpha'}) \tag{40}$$

$$\text{hub} \to \text{IDP} : s_H = \sigma_{\text{hub}}(w \equiv (n', X, B, \Delta, v')) \tag{41}$$

$$\text{IDP} \to \text{hub} : o_{\alpha'} = \mathcal{O}[\chi_{\alpha'}](\alpha', \gamma_{\alpha'}), s_I = \sigma_{\text{IDP}}(w, o_{\alpha'}) \tag{42}$$

$$\text{hub} : (\alpha' \otimes \beta_{\alpha'}) \oplus \gamma_{\alpha'} \overset{?}{=} \delta_{\alpha'}; v = v' \oplus \alpha' \oplus \alpha \tag{43}$$

$$\textbf{Final output} : \text{IDP}[\rho_I, s_H, s_I, w], \text{hub}[\rho_H, s_H, s_I, w, o_{\alpha'}, v] \tag{44}$$

**Fig. 4. Protocol for weak-unlinkability across RPs, with traceability.** Legend: $n$' (session ID); $u$ (user pseudonym at IDP, for interaction with the hub); $r$ (RP identifier at the hub, for interaction with IDP); $\alpha$ and $\alpha'$ ($\oplus$-masks of length $\kappa$); $\beta$ ($\otimes$-randomizer of length $\sigma$); $\gamma$ ($\oplus$-mask of length $\sigma$, used to hide a $\otimes$-randomized value); $\rho_p$ (list of random values selected by $p$); $\chi$ (commitment); $\sigma_p$ (signature by $p$); $\kappa$ and $\sigma$ (computational and statistical security parameters, respectively, e.g., $\kappa = 128$ and $\sigma = 40$; assume $|u| = |r| = |v| = \kappa$); in the $\otimes$-randomization operation, the left and right arguments have respective lengths $\kappa$ and $\sigma$; $\leftarrow^{\$} \mathcal{C}$ (assignment from randomized commitment functionality); $\mathcal{O}[\chi](x)$ (opening of value $x$ from commitment $\chi$ – includes randomness used to commit).

Both parties learn the input maskings ($\Delta$), the $\otimes$-randomizers ($\beta$) of the other party, and the double $\oplus$-masking of $v$ (39).

**Exchange signatures.** At this point, both hub and IDP know the same four commitments (X) and four $\beta$-randomizers (B) (40). The hub signs the concatenation of all commonly known information (41) and sends the signature to the IDP (41). Once the IDP verifies that the signature is correct (not shown), it has assured traceability for itself. The IDP then opens the $\oplus$-mask $\alpha'$ and the respective $\oplus$-mask $\gamma_\alpha$, concatenates the opening (including the randomness necessary to prove a correct opening) to the commonly known information and sends a respective signature to the hub (42). The hub verifies the correctness of the signature (not shown), and that the opened values $(\alpha', \gamma_{\alpha'})$ are correct against the respect masking $\delta_{\alpha'}$ and randomizer $\beta_{\alpha'}$. The hub then removes the two masks ($\alpha$ and $\alpha'$) from the masked output $v'$ to obtain the user pseudonym $v$ for the RP (43).

As final output, each party stores all the random values that may be needed to open the commitments in case of an audit, the exchanged signatures, and all other exchanged values. The hub also stores $v$ (44), which is needed for the subsequent interaction with RP.

**Basic audit example.** Consider an authentication assertion sent from the hub to the RP (22), containing a pseudonym $v$ and a session ID $n$. If an auditor challenges the hub about this transaction, the hub can prove correctness of behavior as follows. First, it shows the respective assertion signed by the IDP (42), which (among other elements) contains a different session ID $n'$, the cryptographic commitments $\chi$ of inputs, the maskings-of-randomizations $\delta$, and the randomizers $\beta$. Second, the hub proves a correct relation between the two session IDs (e.g., based on the traceability solution discussed in Appendix C), and that the IDP is the expected one. Third, the hub opens the $\oplus$-mask $\alpha$, and the respective $\oplus$-mask $\gamma_\alpha$ from $\chi_\alpha$ (29). This allows the auditor to verify that $(\alpha \otimes \beta_\alpha) \oplus \gamma_\alpha = \delta_\alpha$ (35), and then to verify that $v' = v \oplus \alpha \oplus \alpha'$, where $v$ is the value claimed by the hub, and where $\alpha'$, $v'$, $\delta_\alpha$ and $\beta_\alpha$ have all been signed by the IDP.

**Concrete primitives.** The optimal primitives may depend on envisioned audit cases and efficiency tradeoffs. For example, for IDP to prove that in two transactions the pseudonym $u$ is the same (or different), but without revealing the committed pseudonym(s), it may be useful for the commitments ($\mathcal{C}$) to be based on group operations (e.g., Pedersen commitments). On the other hand, the specific commitment of $(\alpha, \gamma_\alpha)$ can be more efficiently based on symmetric primitives because a direct opening is performed within the protocol. Nonetheless, if needed it is also possible to prove in zero-knowledge that certain commitments $(\chi)$ are consistent with the signed maskings

($\delta$) and randomizers ($\beta$) (e.g., using zero knowledge proofs based on garbled circuits [JKO13]). We recommend that envisioned auditability use-cases be made explicit in public documentation by the FCCX and GOV.UK Verify teams.

**Formalizing a solution.** While we claim that the above solution is better than what is proposed by FCCX, we believe that it does not yet reach an adequate level of formalization for implementation. First, integration with other properties, such as with those discussed below, is required, followed by a proof of security. Second, the exact kind of adversarial scenarios one may want to protect against should be further considered. As an example, consider that two different user-pseudonyms ($v_1$ and $v_2$) at an RP are leaked to public information, perhaps due to some unintended data breach related to an audit. Then, the IDP can easily find to which users they correspond, using the following simple procedure. The IDP builds two lists $t_1 = \langle(u, \text{Cipher}_u^{-1}(v_1)) : u \in U\rangle$ and $t_2 = \langle(u, \text{Cipher}_u^{-1}(v_2)) : u \in U\rangle$, where $U$ is the set of all user pseudonyms at the IDP. Then the IDP merges the two lists, sorts the resulting list by the second component of each pair, and finds the two unique pairs that have an equal second component (which is necessarily equal to $r$, the identifier of RP that should be unpredictable to IDP). It follows that the respective first components, $u_1$ and $u_2$, are the exact user pseudonyms at the IDP. From here, the IDP can predict any pseudonym at an RP of a different user at the same RP. This particular issue does not affect any of the properties that we have previously defined (§3), but it does not allow a gracious failure when there is a leakage of pseudonyms. While this particular problem can be resolved by instead having $v = \text{Cipher}_{u \oplus v}(v)$, the example conveys the benefits of further formalization.

### 5.1.2 Strong unlinkability

In the above proposal for weak unlinkability across RPs, the hub still knows the user pseudonym ($v$) sent to the RP, either by choosing it (in FCCX) (step 20 in Fig. 3) or by learning the value decided by the MS (in GOV.UK Verify) (step 19). This allows the hub to link the same user across different transactions associated with the same RP. In a more privacy-preserving solution, the hub would never receive a persistent user pseudonym. This can be solved based on an ephemeral, secret and random key or mask $m$, shared between the IDP and RP and hidden from the hub. It is ephemeral in that it is generated for a one-time use, i.e., once for each transaction. Its secrecy can be derived from an appropriate key-exchange protocol, as suggested in §5.2. Its randomness can be enforced by the hub, via standard and very efficient two-party coin-flipping techniques. Then, the hub and IDP implement an S2PC-based solution, similar in spirit to what we described for weak unlinkability, but with the following differences (ignoring the augmentation for traceability): the IDP also inputs the shared key into the S2PC; the hub inputs into the S2PC an authenticated enciphering of the RP identifier, as received from RP – calculated by the RP using the shared key; the S2PC deciphers the RP identifier, if and only if the shared key inputted by the IDP is the same as the one used by the RP – if the internal verification fails, then the S2PC outputs an error bit and nothing else; the hub then learns a masked and authenticated version of the pseudonym for the RP, instead of the value in clear. Then, the hub sends the output to the RP, who can verifiably open the respective persistent pseudonym. The *traceability* mechanism can be based on the ⊗-then-⊕ techniques already described.

Full unlinkability requires solving also the case across IDPs – this is described in §5.1.3, which in turn takes advantage of the solution just described.

### 5.1.3 Weak unlinkability across IDPs

**Multiple options for connection.** It may be desirable to let a user access the same user account at RP via different accounts at IDPs. This can be achieved after a *matching registration* process at each RP, as follows: (i) first, the user connects using a certain user account at IDP (i.e., a regular transaction); (ii) then, without disconnecting, the user and the RP proceed to a new transaction, with the user reconnecting but through a different account at IDP; (iii) depending on the requisites of the user account at RP, the RP may perform a matching of user-attributes to guarantee, with the adequate level of assurance, that the different connections correspond to the same user; (iv) if the matching is successful, the RP links the two different user pseudonyms into the same local identifier. Thereafter, a transaction through any of the accounts at IDPs leads to the same account at RP. We notice that this procedure breaks *edge unlinkability in regard to changing IDPs*, i.e., the RP knows when the user is changing IDPs.

As an alternative, in FCCX an “account linking” option is provided for a user to link different user accounts at IDPs into a single local account at the hub. It is based on the above registration procedure, but replacing the RP by the hub. The change of IDPs is thus automatically hidden from all RPs, because the user pseudonym that the RP receives from the hub does not vary with the IDP. However, this solution breaks *weak unlinkability across IDPs* and it is further incompatible with weak unlinkability across RPs, as it explicitly allows the hub to link the user to a unique identifier. In other words, in FCCX it is a required tradeoff to allow the user to connect to the same account at RP via several different accounts at IDPs.

In GOV.UK Verify, the MS has the task of matching into a local account the user pseudonym ($u$) and user-attributes (*atts*) received from the IDP (step 16 in Fig. 3). Thus, by default the MSs also break weak unlinkability across IDPs, besides breaking edge unlinkability in regard to RPs that choose the same MS. Even though the MS is instructed to only save a hash version of each $u$, all are linked to a local identifier $v_{\text{Lm}}$. This happens in a compulsory way, without choice by the user.

**Our aim.** We propose that linkability across different IDPs be avoided, while at the same time allowing the user to connect through multiple user accounts at IDPs. Informally, we want the best of both worlds – a user being able to authenticate to the same RP through different accounts at IDPs, such that: (i) the hub cannot link the same user across different transactions; (ii) the RP does not know whether the same or a different IDP is being used; (iii) a single *matching registration* is enough to determine the default behavior for all RPs, but the user can avoid the feature on a per-connection basis; (iv) the user chooses which party to trust with the matching operation, instead of being imposed the hub (in FCCX) or (unknown) MSs chosen by RPs (in GOV.UK Verify).

**Solution description.** We consider a new "identity integration" (II) service. Basically, we propose that the account linking by the hub (in FCCX) and the account matching by MSs (in GOV.UK Verify) be replaced by a corresponding II operation performed by another party optionally chosen by the user. Since the task, involving linking of several user pseudonyms into a single one, is essentially a component of identity and attribute management, it is suited to a credential service provider, such as IDPs and ATPs. For simplicity we describe it here as being a new party on its own, denoted II.

In a voluntary *registration matching* process, the user creates or reconnects to a user account at (a chosen trusted) II to link several (as many as desired) user accounts at IDPs. (We notice that such a kind of registration was already required in FCCX when using *account linking*.) In this special registration transaction, and in spite of mediation by the hub, each IDP is explicitly informed of the identity of the II, so that it can later refer to it. As a result of the registration transaction with each IDP, the II (as a RP) learns a user pseudonym associated with each user account at an IDP, and links it into a local account identifier. However, the II does not need to learn who are the IDPs. Also, the II exchanges a persistent and secret key with each individual user account at IDP.

After a successful registration procedure, automatically if so desired any transaction involving a registered user account at IDP may request the hub for a redirection through the II. Thus, the user may use different accounts at IDP to connect (via regular hub-brokered transactions) to the same user account at a RP. The IDP sends to the hub an assertion that contains a flag related to "account matching" and the identity of the respective II. The hub then communicates directly with the II, as if it were a RP but without redirection through the user.

Initially as a RP, the II learns a user pseudonym; then as an IDP it determines the user pseudonym to use with the hub and it sends a respective assertion to the hub. The hub then completes the transaction by sending a respective assertion to the final RP, via a user redirect. However, based on the proposed solution to strong unlinkability, the hub does not get to learn any user pseudonym. Particularly, the persistent key preshared between the IDP and II allows the IDP to send to II the ephemeral session key necessary for the remainder transaction with the RP. Thus, it follows that, contrary to what happens in FCCX, the hub cannot know if different user accounts at IDPs are linking or not to the same user account at the II (when acting as a RP). The per-connection flexibility can be achieved by the IDP offering to the user, in each authentication, the possibility to decide (e.g., a selectable option) whether or not to use the matching functionality.

In this solution the matching does not need to (but it may) involve attributes, and only occurs if chosen by the user. Compared with GOV.UK Verify, replacing the MS by the II does not disadvantage the RP in terms of authenticity – if the system is upgraded with resilience against impersonation (§5.3) and against linkability (§5.1.1, §5.1.2), the MSs are no longer needed by RPs as an alternative to trusting the hub.

### 5.1.4 Unlinkability against colluding RPs

In FCCX, each RP receives from the hub a different and uncorrelated user pseudonym $v$. However, in GOV.UK Verify such user pseudonym depends only on the MS and on the user pseudonym at the IDP (step 23 in Fig. 3). In other words, $v$ does not change with the RP (assuming the same MS). Thus, two externally colluding RPs (with the same MS) can trivially link the same user in different transactions across the two RPs. This can be avoided by letting $v$ vary pseudo-randomly with the RP. If the MS were to know the RP, which would be detrimental to weak unlinkability, a trivial solution would be to calculate $v$ as a value varying with RP. If the MS does not know the RP, then the user pseudonym could be changed at the hub (e.g., as we propose in §5.1.1 and §5.1.2, also achieving traceability). In GOV.UK Verify, the protocol is geared towards unequivocal identification, not selective disclosure, as it assumes a matching dataset (MDS) of attributes. Thus, this solution in isolation (i.e., varying the user pseudonym with the RP) would not make linkability impossible, but only not easier than what is already possible without brokered identification.

## 5.2 Visibility of attributes

In FCCX and GOV.UK Verify, the attributes integrated in authentication assertions constitute personally identifiable information. In each transaction, the attributes are visible by the hub and (in GOV.UK Verify) the MS, even though the goal of the transaction is to connect the user to the RP. We show that visibility by the hub and/or MSs is completely unnecessary.

**Avoid attribute visibility by the MS.** In GOV.UK Verify, the MS performs a *matching* of user attributes into a local account. We argue that this is avoidable, by contesting the usefulness of the MS. A main argument in favor of the MS-based architecture would seemingly be to allow each RP to choose which MS to trust, instead of having to trust the hub. However, such argument is void since, as shown in §5.3: (i) in GOV.UK Verify a malicious hub can still perform impersonation attacks; (ii) it is possible to achieve resilience against a malicious hub even without using a MS. Second, user matching based on the *matching dataset* of user attributes is not required if the IDP already possesses those attributes, and is otherwise possible via attribute enrichment aided by ATPs. So, the user, rather than the RP, should control which party enables the matching, and only *if* and *when* needed (instead of in every transaction), as already argued in §5.1.3. This achieves the best of both worlds: RPs do not have to trust the hub; users can choose which IDPs to trust, preventing linkability by the MSs.

**Share an ephemeral key.** To avoid attribute visibility by the hub, the IDP can encrypt the attributes under a cryptographic key known by the RP. However, to ensure edge unlinkability, the IDP must not know the (persistent) public key of the RP, lest the IDP could infer which service a user is about to access. Thus, the RP and IDP should anonymously share a random ephemeral key. A Diffie-Hellman Key-Exchange (DHKE) [DH76] mediated by the hub would provide a solution in the honest-but-curious setting. However, we want a solution resilient to an active man-in-the-middle (MITM) attack performed by a malicious hub. A specific difficulty is that the RP and IDP are anonymous vis-a-vis each other, making it difficult to authenticate messages between them. This can be overcome by an *auxiliary-channel*-DHKE type of protocol [NLJ08], which involves exchanging between RP and IDP a short value that is unpredictable by the hub in a single trial.

To achieve the above mentioned solution, the RP could initially (step 1 in Fig. 3) show a short random session identifier (a “PIN” of 8-10 characters) to the user, who would then input it back (e.g., through copy-and-paste) during her authentication with the IDP, along with her login credentials (step 9 in Fig. 3). A MITM attack would succeed only in the unlikely case the hub correctly guesses the PIN on a first try, and would fail and be detected in any other case; the resulting exchanged key could still be as large as needed. Another way, additionally ensuring edge unlinkability even in presence of a malicious RP, is to have the user choose the random PIN. This could be facilitated by an embedded functionality in web browsers, or by separate software or hardware (e.g., similar to what is used in certain two-factor authentication mechanisms), potentially on a different medium. In §5.3 we show that, to prevent certain impersonation attacks, when a malicious hub colludes with a malicious relying party, the PIN should ideally be obtained jointly by the user and the RP, e.g., using an efficient two-party coin-flipping protocol.

**Usability.** A usability evaluation, which we advocate even without PINs, would be useful to determine how to best integrate user interaction into the transaction protocol. For a user, the complexity of inserting a PIN is likely comparable to the complexity already required to authenticate to the IDP via a username and password. The burden of manual intervention by the user would be comparable to that required to solve a typical CAPTCHA [vABHL03], e.g., inserting a short code into an input field (upon some mental work) – a security measure that would already make sense at the RP side. Alternatively, the whole process of copy-and-paste and/or random sampling could be made seamless if the design constraints allowed user-side software (optionally trusted by the user) to automate some of these actions.

**Integrate attributes.** Once a key is shared between IDP and RP, they can resolve different types of requests: transmission of (a group of) attributes; *secure comparison* of attributes; or any of the previous but associated with a certain predicate of the attributes. Secure comparison can be achieved by each party (RP and IDP) hashing the attributes, then enciphering the hash (using the ephemeral random key) and then sending the resulting ciphertext to the hub, who then sends the result of comparison to RP. In the case of transmission of attributes, the IDP can simply send the attribute values in encrypted form, even though this hinders the enforcement of edge unlinkability. Authenticated encryption can be used to ensure confidentiality and integrity, in spite of hub mediation. Alternatively, this may be reduced to *secure comparison* if the user serves as an auxiliary channel, inputting its own attributes in RP and then requesting a secure comparison to be performed. There may also be value in hiding from the hub some details of the attribute request, e.g., the kind of predicate being compared.

## 5.3 Impersonation by a malicious hub

We show four attacks where a malicious hub violates authenticity by successfully impersonating a legitimate user, in a variety of contexts.

**Impersonation at intended RP.** A compromised hub proceeds with the protocol until building the authentication assertion required for the RP, but then impersonates the honest user to conclude the protocol with the RP (Steps 22 and 23 in Fig. 3). The attack succeeds if the RP has not established a shared secret with the honest user, to verify that the user did not change during the execution of the protocol. To foil this attack, the RP may set a fresh secret cookie on the user agent in the first step of the transaction, similar to a cross-site request forgery token [BJM08]. The hub would not have access to this cookie, and later in the protocol the RP would confirm that the cookie is held by the user. Unfortunately, the FCCX and GOV.UK Verify documentation make no mention of this type of measure. Regardless, the next three impersonation attacks succeed even assuming that this cookie protection is in place.

**Impersonation at any RP, without user authentication at IDP.** In FCCX, access to a user account at an RP depends only on an assertion signed by the hub with a respective user pseudonym ($v$) (step 22 in Fig. 3) and attributes. Thus, once a malicious hub has brokered access from a user to an RP, it can arbitrarily replay the access in the future, without even involving an IDP. The same attack can be performed in GOV.UK Verify if a malicious hub and MS collude (step 23). The attack might be detected only a posteriori, if the RP gives the respective assertion (signed by the hub and/or MS) to an auditor and the auditor requests from the hub (or MS) the respective assertion signed by IDP, which does not exist. The attack can be foiled by preventing the hub (and MS) from learning the user pseudonym at any given RP, which is precisely what our solution for strong unlinkability (§5.1.2) achieves.

**Impersonation at any RP, upon user authentication at IDP.** After an honest user initiates a transaction with RP$_1$, a maliciously compromised hub receives an authentication request with ID $n_1$ (step 4 in Fig. 3). Then, impersonating the user, this hub starts a new transaction with RP$_2$ (which may or may not be RP$_1$), obtaining a new request ID $n_2$. The hub then sends to the IDP an authentication request with ID $n_2$ (instead of $n_1$) (step 8). In return, it receives an authentication assertion signed by the IDP (step 12), associated with $n_2$ and with $u$. Then, the hub simply continues the transaction that it initiated, impersonating the victim user (involving MS, in GOV.UK Verify) until it gains access to RP$_2$ as the impersonated user. The attack does not break traceability in FCCX or GOV.UK Verify, because the authentication ID $n_2$ is signed by all parties, and goes undetected by MS and RP$_2$, who receive expected verifiable signed authentication assertions. The legitimate user only experiences an aborted execution, possibly camouflaged as a network/connection error, and without visibility over $n_1$ and $n_2$. The attack is possible due to insufficient binding between the user request at RP and the user authentication at IDP. The user has no guarantee that the relation with the intended RP will be maintained. This attack too can be defeated by our proposal to enforce strong unlinkability, with the user transmitting a (random) PIN between RP and IDP, which allows RP and IDP to share a random key. Then, the originating RP will extract a valid pseudonym only if the shared key is associated with the intended user account at the IDP.

**Impersonation at unintended RP$_2$, if a malicious hub colludes with the intended (but malicious) RP$_1$.** Assuming the solution to the previous impersonation attacks (user-transmitted PIN) is deployed, there remains a possible attack if (i) the malicious hub colludes with RP$_1$ with which the user wants to authenticate, and (ii) the user selects the PIN herself. Indeed, the (malicious) RP$_1$ can inform the malicious hub of the user PIN. Then, the hub can impersonate the user requesting access to a different (honest) RP$_2$ using the same PIN, for a transaction with a new random authentication ID $n_2$. The hub then leads the user to authenticate into the IDP, using authentication ID $n_2$ instead of $n_1$. The user will then provide the IDP with the expected PIN. Knowing the PIN, the malicious hub can perform a successful active MITM attack, sharing one key directly with IDP and (possibly another) with RP$_2$, and eventually gaining access to RP$_2$ as the impersonated user. In FCCX, the hub gains further ability to impersonate a user at any later time, even without involving the IDP, because it learns the user pseudonym $v$. To thwart the described attack, the PIN may be decided by a coin-flip between user and RP, ensuring it is random even if either RP or the user (e.g., in case of impersonation) are malicious. Alternatively, if the user is willing to forgo edge unlinkability against a malicious RP (e.g., due to usability reasons), then the PIN can simply be chosen by the RP (and still transmitted to the IDP by the user).

## 5.4 Traceability and forensics

**Traceability.** The signatures exchanged between parties in FCCX and GOV.UK Verify provide some level of traceability if the hub is honest. If questioned by an auditor about a certain authentication assertion or request, the hub can show a related assertion or request (assuming respective logs are recorded). However, for the same request from RP or same assertion from IDP, a malicious hub may produce several requests and assertions. For example: (i) in GOV.UK Verify, the hub could undetectably send the assertion to two different MSs, to illegitimately obtain two user identifiers; (ii) in FCCX, the hub could undetectably produce assertions for two different RPs; (iii) in FCCX and GOV.UK Verify the hub could collude with a rogue IDP, to obtain a respective assertion, besides the legitimate one. Later, in a limited audit the malicious hub could use the most convenient justification, while hiding the legitimate and/or other illegitimate ones. We argue that this can be improved with a property of *one-to-one* traceability. The intuition is that each party commits to the details of the next action, before receiving the "justification" (i.e., the signed material) referring to the previous action. The commitments need not be opened during the transaction, but only in case of auditing. We have already sketched in §5.1.1.1 how to achieve this in connection with weak unlinkability, with respect to pseudonyms. In Appendix C we give another example related to session IDs.

**Selective forensic disclosure.** The coarse-grained nuance is trivially possible by the weak and strong unlinkability property that we have proposed. For that, a compelled IDP just needs to let the hub know of all transactions associated with the same user account at IDP. While this breaks weak unlinkability for the user, it preserves the privacy of other users and so it is already strictly better in comparison with FCCX and GOV.UK Verify, where selectiveness of forensic disclosure is not possible (as linkable identifiers are leaked to the hub by default in all transactions). The fine-grained nuance is possible via a different procedure, based on a transaction flagged as a forensic investigation. Given a pinpointed transaction with (IDP, user, $\text{RP}_1$), and a targeted $\text{RP}_2$, the hub initiates a forensic transaction between $\text{RP}_2$ and IDP. This is a regular transaction but with two main differences. First, since the IDP is collaborative, it interacts with the hub as if the actual user (defined by $u$) had authenticated – as a result (and assuming the solution for strong unlinkability) the RP learns (but the hub does not) the respective user pseudonym at the RP. Second, the RP receives information that this is a forensic transaction – the information is signaled through the IDP, independently of the hub, in order to prevent a malicious hub from actually impersonating the user. This allows RP to control whether or not (depending on the subpoena) to give the hub access to the internal user account. Then, a collaborative RP may inform the hub about the past transactions (i.e., their session IDs) involving the same user account.

# 6 Concluding remarks

We have evidenced severe privacy and security problems in FCCX and GOV.UK Verify and have shown feasible solutions to address them. Passively, the hub is able to profile all users in respect to their interactions across different service providers. If compromised, the hub can even actively impersonate users to gain access to their accounts (and the associated private data) at service providers. This represents a serious danger to citizen privacy and, more generally, to civil liberties. The described vulnerabilities are exploitable and could lead to undetected mass surveillance, completely at odds with the views of the research community [IAC14] whose scientific advances enable feasible solutions that are more private and secure.

Based on the findings presented in this paper, we believe that a security review should lead to fundamental structural adjustments in the interest of privacy and security. It is clear that the FCCX and GOV.UK Verify do not adequately consider the need for resilience against a compromised hub and fail to address plausible threats.

We have described solutions to the main problems we identified. However, a comprehensive solution for brokered identification would require greater formalization and we hope this paper serves as a call for more research. One would need a design specification and proper requirements, followed by a fully specified, unambiguous protocol accompanied by a proof of security. As a first step, we strongly recommend that a formal framework for brokered authentication be devised, perhaps based on the ideal/real simulation paradigm (e.g., [Can01]). Such a framework would integrate all the security, privacy and auditability properties at stake, while considering an adversarial model in which any party, including the hub, may be compromised and/or collude with other parties.

# Acknowledgments

Luís Brandão is a Ph.D. student at CMU-ECE and FCUL-DI supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under Grant SFRH/BD/33770/2009. Nicolas Christin was also partially supported by the CMU|Portugal program. George Danezis is supported by EPSRC Grant EP/M013286/1 on "Strengthening anonymity in messaging systems." George Danezis thanks the UK Government Digital Services for conversations related to the UK system (which does not imply they endorse our conclusions). The authors thank Rene Peralta for significant contributions to this work, Erman Ayday for shepherding the final revision of this paper, and the anonymous reviewers for valuable feedback that contributed to greatly improve the writing.

# References

[AMPR14] A. Afshar, P. Mohassel, B. Pinkas, and B. Riva. Non-Interactive Secure Computation Based on Cut-and-Choose. In P. Nguyen and E. Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, vol. 8441 of *LNCS*, pages 387–404. Springer Berlin Heidelberg, 2014.

[ARS$^+$15] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, vol. 9056 of *LNCS*, pages 430–454. Springer Berlin Heidelberg, 2015.

[BBF$^+$13] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon. XML Signature Syntax and Processing Version 2.0. *W3C Working Group Note*, April 11, 2013.

[BD11] P. Beynon-Davies. The UK national identity card. *Journal of Information Technology Teaching Cases*, 1(1):12–21, 03 2011.

[BJM08] A. Barth, C. Jackson, and J. C. Mitchell. Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 75–88. ACM, 2008.

[Bra13] L. T. A. N. Brandão. Secure Two-Party Computation with Reusable Bit-Commitments, via a Cut-and-Choose with Forge-and-Lose Technique. In K. Sako and P. Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013*, vol. 8270 of *LNCS*, pages 441–463. Springer Berlin Heidelberg, 2013.

[Bri15] Bristol Cryptography Group. Circuits of Basic Functions Suitable For MPC and FHE. http://www.cs.bris.ac.uk/Research/CryptographySecurity/MPC/, Accessed February 2015.

[Can01] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *FOCS 2001*, pages 136–145, 2001.

[CES12] CESG and NTAIA and Cabinet Office. Good Practice Guide No. 43 – Requirements for Secure Delivery of Online Public Services, December 2012. PDF file (46 pages) – gov.uk website. SHA256: b765484870bac4e5 d3ecf12a6cc72923 69870bbac41b48fb 55eb8163006706ab.

[CES14] CESG and NTAIA and Cabinet Office. Good Practice Guide No. 45 – Identity Proofing and Verification of an Individual, July, 2014. PDF file (32 pages) – gov.uk website. SHA256: 99870cebd50ab893 ad5bb25a96f9b33c f58a19e2472d63c0 71794b104fd15b5b.

[Cyb11] Cyber-Auth DG Committee – Canada. Cyber Authentication Technology Solutions Interface Architecture and Specification Version 2.0: Deployment Profile, March 25, 2011. PDF file (53 pages) – kantarainitiative.org website. SHA256: 2faf71b0d92d83e3 18208d6676743ad1 e624025fc14efdb3 9f19ec750d1cf6ed.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[DLT14] I. Damgård, R. Lauritsen, and T. Toft. An Empirical Study and Some Improvements of the MiniMac Protocol for Secure Computation. In M. Abdalla and R. De Prisco, editors, *Security and Cryptography for Networks*, vol. 8642 of *LNCS*, pages 398–415. Springer International Publishing, 2014.

[Eur10] European Court of Human Rights. European Convention on Human Rights, Entry into force on June 2010. (As amended by Protocols No. 11 and 14; supplemented by Protocols Nos. 1,4,6,7 and 13.) PDF file (30 pages) – echr.coe.int website. SHA256: 4c65e157638088a7 a0905352fbed358a ca0f79564a8dfc06 d3246ee2cd7becfa.

[FF11] Federal Chief Information Officers Council and Federal Enterprise Architecture. Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance – Version 2.0, December 2, 2011. PDF file (478 pages) – idmanagement.gov website. SHA256: f4b91795175b2a73 1f3b29c32d5e5fef eaa1cb97d0cec3c2 522f4fe8c51a5bdc.

[FJN14] T. K. Frederiksen, T. P. Jakobsen, and J. B. Nielsen. Faster Maliciously Secure Two-Party Computation Using the GPU. In M. Abdalla and R. De Prisco, editors, *Security and Cryptography for Networks*, vol. 8642 of *LNCS*, pages 358–379. Springer International Publishing, 2014.

[Gen14] General Services Administration. Solicitation Number QTA0014AWA3005 – Identity Services Support. Federal Business Opportunities (FedBizOpps), Updated on August 19, 2014. ZIP files – fbo.gov website: "Amendment_1.zip" (Aug 06, 2014) SHA256: 08a6a9f4b59f9ce2 466c1a90049b2498 1385386aa7df0a4f 8b2db74c50848881. "Amendment_2.zip" (Aug 19, 2014) SHA256: 0c078fe1266a1f3b 822fbd9a35c212e8 717292331a158c27 f28af56764267fbb.

[HKK$^+$14] Y. Huang, J. Katz, V. Kolesnikov, R. Kumaresan, and A. Malozemoff. Amortizing Garbled Circuits. In J. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, vol. 8617 of *LNCS*, pages 458–475. Springer Berlin Heidelberg, 2014.

[IAC14] IACR Members Meeting held at Eurocrypt 2014. IACR Statement On Mass Surveillance – “Copenhagen Resolution”, May 14, 2014.

[Ide13a] Identity Assurance Programme. Identity Assurance Hub Service SAML 2.0 Profile v1.1a, September 11, 2013. PDF file (36 pages) – gov.uk website. SHA256: 1b5770b030526414 c64d7f55d6dd4340 c5cdf136355837b3 29370aab1bcde35c.

[Ide13b] Identity Assurance Programme. Identity Assurance Hub Service Profile – SAML Attributes v1.1a, September 11, 2013. PDF file (12 pages) – gov.uk website. SHA256: e24773e9436eaf38 afc9a51d8a069e2d 9314e81dd1c7363a 433365164dd2acd4.

[IDS02] T. Imamura, B. Dillaway, and E. Simon. XML Encryption Syntax and Processing. *W3C recommendation*, December 10, 2002.

[Int08] Internet Engineering Task Force (IETF) – Network Working Group. Request for Comments: 5246 – The Transport Layer Security (TLS) Protocol (Version 1.2), August 2008. RFC5246. See also: the Errata; and RFC6176 from March 2011.

[ISO13] ISO/IEC. Anonymous Digital Signatures. *Information technology – Security techniques*, ISO/IEC 20008-1:2013, 2013.

[JKO13] M. Jawurek, F. Kerschbaum, and C. Orlandi. Zero-knowledge Using Garbled Circuits: How to Prove Non-algebraic Statements Efficiently. In *Proc. 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 955–966. ACM New York, NY, USA, 2013.

[Joh12] A. John. Challenges in Operationalizing Privacy in Identity Federations – Part 1, Part 2 and Part 3. IDMGOV Info – U.S. FICAM program, 2012. Blog posts – info.idmanagement.gov website.

[LR14] Y. Lindell and B. Riva. Cut-and-Choose Yao-Based Secure Computation in the Online/Offline and Batch Settings. In J. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, vol. 8617 of *LNCS*, pages 476–494. Springer Berlin Heidelberg, 2014.

[NIS13] NIST – Computer Security. NIST Special Publication 800-130– A Framework for Designing Cryptographic Key Management Systems, August 2013. SHA256: 937d2f89c476a055 46193d7801f8f0d2 419844c81912a1aa 2c74cf38ce860dc4.

[NLJ08] D. K. Nilsson, U. E. Larson, and E. Jonsson. Auxiliary Channel Diffie-Hellman Encrypted Key-exchange Authentication. In *Proc. 5th Int. Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, QShine, pages 18:1–18:8. ICST, 2008.

[NST13] NSTIC National Program Office. NSTIC Requirements Document, September 10, 2013. Xlsx spreadsheet – idecosystem.org website. SHA256: e359eea01c79b415 0048628273ec4f62 5faaa8c397c187aa 998b09128741dfd2.

[OAS05] OASIS. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard – March 15, 2005. PDF file (86 pages) – oasis-open.org website. SHA256: dc0890f88bfe862c b0bedb03491cc41c 9b39672bed76e2dc 161aea472b0468ab.

[Ope] OpenID Foundation. OpenID Connect. Available online at the openid.net website.

[Pri14] Privacy and Consumer Advisory Group (PCAG). Identity Assurance Principles – v3.1 (for publication), 2014. PDF file (12 pages) – gov.uk website. SHA256: 3f3a08a84e35f6e2 3ebc9f691f1816d1 5c303ce10ecd1805 3be2eb7a08727bd5.

[The11] The White House. National Strategy for Trusted Identities in Cyberspace – Enhancing Online Choice, Efficiency, Security, and Privacy, April 2011. PDF file (52 pages) – whitehouse.gov website. SHA256: 2dfcb30eca1bf3e0 2b91addaaabde5b9 bfc463bfd270ba43 fc478af0bfa621e3.

[Uni13] United States Postal Service. Solicitation Number 1B-13-A-0003 – Federal Cloud Credential Exchange (FCCX). Federal Business Opportunities (FedBizOpps), 2013. ZIP files – fbo.gov website: "RFP_Documents.zip" (January 10) SHA256: 72c68c9fe1be32da 2c586edab757d933 385811fada90c0b9 57ee9a6f3ecfb13a; "Amendment_5.zip" (January 28) SHA256: a61df077ee03fb3c 56d6ded65eb5058e 2a1db13deadb2ba7 961a237d81e6213b.

[USP14] USPS – Information Security and Privacy Advisory Board. FCCX Briefing, June 13, 2014. PDF presentation (24 slides) – csrc.nist.gov website. SHA256: 144135e4bb08419f d690446fcb12e447 854620780e93a1fb 74b9afa6249c0c99.

[vABHL03] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, vol. 2656 of *LNCS*, pages 294–311. Springer Berlin Heidelberg, 2003.

[Wre12] S. Wreyford. Identity Assurance goes to Washington. GDS Blog, May 29, 2012. Blog posts – gds.blog.gov.uk website.

# A Integration of attributes

**Supported attributes.** In Fig. 3, we left implicit in the authentication requests (steps 4, 6 and 7) what are the requested attributes. In GOV.UK Verify, there is a set of default attributes, denoted *matching dataset* (MDS), which is explicitly required by MS, so we assume that they will always be provided (if available) by the IDP. They are composed of the first-name, surname, middle names, date of birth, gender, current and previous address [Ide13b]. In the "current FCCX flow" [USP14], attributes are integrated in the authentication assertion sent by IDP. The respective "Identity Services Support" solicitation states that "CSPs are expected to support returning predetermined attributes, based on the attributes specified in the SAML metadata." This may involve "dynamic combinations of attributes (attribute request groups) made of individual attributes (or parsed portion of an attribute, e.g., the last four digits of social-security number)" if requested by the RP [Gen14]. The possible *attribute groupings* are still to be defined. The required supported attributes are the legal first and last name, middle name or initial, current address (parsed and full), date of birth (parsed and full), social security number (parsed and full), email address.

We recommend that a good solution should: allow transactions without attributes; require informed user consent about each request, after making explicit which attributes are involved; achieve *traceability* in respect to attributes, e.g., so that hub can prove that attributes requested to IDP are not more than those required by the RP. Also, in a privacy preserving solution, where the replied attribute values are hidden from the hub (this is not the case in FCCX or GOV.UK Verify), there may be value in hiding from the hub the type of attribute request (e.g., which attributes are being requested).

**Non-supported types of transactions.** NSTIC foresees several types of transactions, with different nuances of anonymity and identification [NST13, Req. 12]. For a transaction to be *anonymous with validated attributes*, the RP does not receive a user pseudonym but verifies some attributes; e.g., it learns that the age of the user is within a certain range, but is not able to link different transactions to the same user. For a transaction to be *pseudonymous without attributes*, the RP always learns the same pseudonym for the same user but does not learn any user attribute. At present, it appears that these more privacy-preserving types of transactions are not supported in FCCX and GOV.UK Verify. We find a hint in FCCX for pseudonymous identification [Gen14, Q&A 2.2], but only subsequent to an initial identification with attributes, i.e, when RP can already associate the pseudonym with attributes [USP14].

**Attribute enrichment.** The case of *attribute enrichment* corresponds to additional ATPs complementing the attribute values that the original IDP may (or may not) provide. In GOV.UK Verify, this is described [Ide13a, Steps 9 to 19] but it is informed that it will not be implemented in the current service. In FCCX, this is illustrated by a use-case [Uni13, Use-Case 7b] and other high-level requirements, but some aspects are undefined. In both cases, the hub determines who are the appropriate ATP(s) and then requests and obtains the respective user attributes – it is not clear how the ATP is located, and it seems that the user is not involved in the process. We propose instead that the hub (or the IDP) should ask the user to choose an ATP, in the same way that it asks to choose an IDP, i.e., by means of a selector list and then followed by authentication. A needed technicality is ensuring that the accounts at the several IDP/ATPs correspond to the same user (e.g., via user matching, as suggested in §5.1.3) while solving the problem of unlinkability across IDPs. There are other complexities that may be associated with attributes:

- "Account name mapping" may be required to uniformize the labels given to each attribute, if different parties (e.g., RP, IDP) use different labels – this can be resolved without involving the values of the respective attributes;
- "Consolidation of attributes" may be required when the reply to send to RP derives from a composition (e.g., involving conjunctions and disjunctions) of attributes from different ATPs – some of these cases can be directly solved with clever techniques, without requiring visibility of attribute values. If there are remaining cases where consolidation is difficult, we recommend that the user is allowed to choose a trusted IDP/ATP (or II, as in §5.1.3) to perform such operation, instead of allowing attribute visibility to the hub.

**Consent for attributes.** The matter of user consent concerning attributes was not included in the described protocol (Fig. 3). For FCCX, the recent "Identity services support" solicitation requires that the user sees the actual values and gives opt-in consent, before transmission [Gen14]. However, it is not clear if consent will be asked before the attributes are actually requested to the ATP and before they pass in clear through the hub. In GOV.UK Verify, the user consent is requested if the requested attributes are beyond the default MDS [Ide13a, Step 12], but it is not clear if the user sees the attribute values. Also, in GOV.UK Verify the user does not have any control over the attribute values kept by the MS. In both systems, it seems that the hub interfaces directly with the ATPs.

A a way to improve the privacy of the user, we suggest that the user be given the opportunity to give informed consent in two stages. First, about the type of request, either when contacting the RP or when asked to select an IDP. Second, upon confirming the specific attributes values, at the IDP. For example, once logging into the IDP, the user could visualize the attribute values and reconfirm (i.e., decide whether or not to allow) their transmission/integration.

We advocate that the same type of consent should be required when involving ATPs other than the IDP. Since it seems that in FCCX and GOV.UK Verify the user is not involved in selecting and/or authenticating to the ATP, here the consent could have to be mediated by the hub. However, we notice that it would be a privacy shortcoming to allow the user to give consent about attribute values only after the hub has already seen such values. This is an incentive to consider a solution where the user can choose another party (i.e., not the hub) with whom to trust the process of attribute enrichment, e.g., possibly the IDP, or a single ATP. We also advocate that, whenever technically possible (namely regarding usability restrictions on the user), the user be given opportunity to choose the type of attribute integration, e.g., secure comparison instead of direct transmission. Secure comparison might involve the user passing to the RP, e.g., via copy-and-paste, her own attribute values, so that the RP can then perform a secure comparison with the IDP and/or ATPs, intermediated by a sightless hub.

# B Unlinkability of other identifiers

Side-channels or covert channels can be used with or without intention to leak identifiers that allow some types of linkability.

**Weak unlinkability vs. user-agent identifiers.** Weak and strong unlinkability are defeated if the hub is able to use side-channel information to link the user across different transactions. For example, on a typical usage a user may keep invariant her own IP-address, and/or other information leaked about the user-agent (e.g., identifiers of the web-browser and plugins). The techniques to reduce such side-channel information are outside of the scope of this paper (e.g., connecting through an anonymized network, and using software that shields the amount of leaked information). Nonetheless, we advocate that such concerns should be made explicit in regard to the specification of the records kept by the hub, and that the users should be informed of the respective potential for linkability.

**Edge unlinkability vs. attributes.** Several side-channels from IDP to RP also exist, e.g., in the encoding of attributes, contextual records, and respective complex XML structures. We do not find any suggestion in FCCX of GOV.UK Verify about normalization, or active sanitization by the hub or MS, to avoid such side (or even covert) channels. They can for example be avoided by enforcing secure comparison of attributes, instead of direct transmission. If the RP already knows the candidate attribute values then the hub can mediate a secure comparison, by comparing randomized hiding commitments of the attributes, as submitted separately by IDP and RP. A random value shared between IDP and RP, and kept secret from the hub (see §5.2) would determine the randomization.

**Edge unlinkability vs. session ID.** SAML specifies that the authentication request ID $n$ must be unique [OAS05, §1.3.4], but does not specify that it cannot be linkable to the issuer. Thus, if it is an invariant across a transaction, i.e., the same at RP and IDP (as required in GOV.UK Verify), then it can be trivially used to leak to the IDP who the RP is, thus breaking edge unlinkability. This could happen even in compliant implementations, e.g., if it is a counter (likely to be different across RPs and thus acting as a pseudonym), or even a concatenation of the RP identifier and a counter. A malicious RP could further define $n$ as an enciphering of a randomized version of its own public identifier, thus revealing itself to an IDP possessing the appropriate key; no auditing by the hub (oblivious to the key) would distinguish this request ID from a genuinely random one. One solution against this is to use an efficient coin-flipping protocol between the hub and RP, to enforce a random session ID. Alternatively, and additionally avoiding linkage through session IDs even if comparing the databases of the IDP and RP, the hub may use with IDP an authentication request ID ($n$') different from the one ($n$) received from RP. Care is needed in this case to ensure traceability (see §C).

# C Traceability of session ID

Traceability allows auditability of transaction steps, by allowing each party in isolation (RP, IDP, hub, MS) to prove, to an auditor, that its actions were justified. For example, a signed authentication request sent from the hub to an IDP should be justifiable only by a respective signed request received from a RP. Here, we consider only the case of session IDs. A complete solution to brokered identification should integrate several privacy and security properties, so this discussion serves only as an example of the spirit in which to achieve traceability. A final solution should undergo greater formalization and analysis.

Below, we assume that the session ID ($n'$) used for interaction between the hub and IDP is different from the one ($n$) used for interaction between the hub and RP in the same transaction. (We do not know if this is the case in the actual FCCX system.) For example, this may be desirable to simultaneously prevent the session ID from being used as a covert-channel from RP to IDP, and from being a common identifier at the two edges. While the covert channel could be avoided by deciding the session ID via a two-party coin-flipping between the hub and the RP, its commonness at both edges can only be avoided by actually transforming the session ID. However, if the transformation imposed by the hub is arbitrary, then a maliciously compromised hub could produce different plausible justifications for each step of a transaction. Such a hub could initiate several paths of execution and later (during an audit) only reveal the most convenient one. For example, if a signed request sent to IDP were audited, the hub could associate with it any signed request received from any RP. (We are ignoring here other metadata; e.g., if timestamps are present then they need to remain consistent across the steps of the transaction.)

**Intuition.** Let $n$ be the request ID upon a successful initial interaction between the RP and the hub. We want a mechanism such that a malicious hub later interacting with the same RP, or a honest hub later interacting with the same but malicious RP, is not able to obtain a signed request with the same ID. Furthermore, if later a malicious hub and a different and malicious RP collude to produce a signed request with the same ID, then it should not be possible, in an audit, to validate such request in connection with this other RP. These properties can be achieved by deriving the request ID from two random contributions, one from the hub and another from the RP, and also from an identifier of the RP, in a collision-resistant manner. One can also easily embed an association with an identifier of the IDP, by having the hub learn in advance which IDP the user selects. In such a case, an auditor can only successfully verify the session ID in association with a single pair (IDP, RP).

Then, let $n'$ be the request ID used by the hub in the subsequent request sent to the IDP. The new request ID should be derivable unequivocally by the hub from the previous request ID $n$ and its associated secret information. However, it should not be decidable by the RP (who also knows $n$), so that it is not linkable upon collusion of IDP and RP. This can be achieved by letting $n'$ depend on information committed by $n$, provable by the hub and unknown by the RP.

A proof-of-concept of how to achieve this is given in Fig. 5. We emphasize that this is not an optimized protocol and better solutions may exist, namely when integrating other metadata, a set of intended audit actions and remaining aspects of the overall brokered identification protocol.

**Description.** Initially, the RP selects an auxiliary session ID $n_{\mathrm{R}}$ for communication with the hub (45). The RP then sends encrypted to the hub, via a user redirection, this session ID along with a signature (46). We leave implicit the metadata necessary for the hub to understand which IDPs might be able to fulfill the request. At this point, the hub interacts with the user to select an IDP (47). Then, the hub selects a random *transformer* element $n_{\mathrm{H}}$ (48), which will later be helpful to *transform* a request ID from the RP into a request ID to the IDP. The hub then computes: a commitment $\chi_{\mathrm{H}}$ to the triplet composed of the RP identifier, the IDP identifier and the transformer element (49); and the request ID $n$ for the RP as a collision-resistant hash of the triplet composed of the identifier of the RP, the auxiliary session ID $n_{\mathrm{R}}$ decided by the RP and the commitment computed by the hub (50). The hub then sends directly to the RP (i.e., without user redirection) a message with the request ID $n$ and the respective components of the pre-image (51). The RP verifies that the received request ID is consistent with the hash of the respectively received preimage (52) and also (not shown) that the RP identifier is correct. Finally, the RP sends to the hub an actual authentication request $\{n\}$, properly formatted for logging purposes, including a respective signature with the accepted request ID $n$ (53). The hub verifies (not shown) that the request ID is as expected and that the signature is correct.

The hub then concatenates the request ID $n$ from the RP with the secret transformer $n_{\mathrm{H}}$, and calculates the respective collision-resistant hash, thus obtaining a value $r$ that is unpredictable by RP (54). We assume here a random oracle model, i.e., that when the pre-image is unpredictable, the hash output is pseudo-random. The hub then concatenates this value to an identifier of the IDP and computes again a collision-resistant hash to obtain the new request ID to use with the IDP (55). Finally, the IDP sends to the IDP, via a user-redirection, the signed request adjusted as need be, namely using the new request ID $n'$, and disclosing the preimage containing the IDP identifier (56). Thus, the IDP can verify (not shown) that it is the only valid recipient of this ID.

$$\mathrm{RP} : n_{\mathrm{R}} \xleftarrow{\$} \{0,1\}^{|\mathrm{authN\ ID}|} \quad \text{(auxiliary session ID)} \qquad (45)$$

$$\mathrm{RP} \rightsquigarrow \mathrm{hub} : E_{\mathrm{hub}}\left(n_{\mathrm{R}}, \sigma_{\mathrm{RP}}(n_{\mathrm{R}})\right) \qquad (46)$$

$$\mathrm{hub} \leftrightarrow \mathrm{user} : \ \text{Select IDP} \qquad (47)$$

$$\mathrm{hub} : n_{\mathrm{H}} \xleftarrow{\$} \{0,1\}^{|\mathrm{authN\ ID}|} \quad \text{(select random transformer)} \qquad (48)$$

$$\mathrm{hub} : \chi_{\mathrm{H}} \xleftarrow{\$} \mathcal{C}(\mathrm{RP}, \mathrm{IDP}, n_{\mathrm{H}}) \ \left(\substack{\text{commitment to edges} \\ \text{and transformer}}\right) \qquad (49)$$

$$\mathrm{hub} : n = \text{CR-Hash}(\mathrm{RP}, n_{\mathrm{R}}, \chi_{\mathrm{H}}) \ \text{(request ID for the RP)} \qquad (50)$$

$$\mathrm{hub} \rightarrow \mathrm{RP} : E_{\mathrm{RP}}\left(n_{\mathrm{R}}, n, \mathrm{RP}, \chi_{\mathrm{H}}\right) \qquad (51)$$

$$\mathrm{RP} : \text{Verify CR-Hash}(\mathrm{RP}, n_{\mathrm{R}}, \chi_{\mathrm{H}}) \overset{?}{=} n \qquad (52)$$

$$\mathrm{RP} \rightarrow \mathrm{hub} : E_{\mathrm{hub}}\left(n_{\mathrm{R}}, \{n\}, \sigma_{\mathrm{RP}}(\{n\})\right) \qquad (53)$$

$$\mathrm{hub} : r = \text{CR-Hash}(n, n_{\mathrm{H}}) \qquad (54)$$

$$\mathrm{hub} : n' = \text{CR-Hash}(\mathrm{IDP}, r) \ \text{(request ID for the IDP)} \qquad (55)$$

$$\mathrm{hub} \rightsquigarrow \mathrm{IDP} : E_{\mathrm{IDP}}(\{n'\}, \sigma_{\mathrm{hub}}(\{n'\}), \mathrm{IDP}, r) \qquad (56)$$

... (protocol continues)

**Fig. 5. Achieving traceability with changing authentication request ID.** Note: this example only shows the component RP→hub→IDP. See legends of Fig. 3 and Fig. 4.

From a user-experience perspective, the protocol retains the same flow as in FCCX or GOV.UK Verify, because no further redirections were needed. The user is at most subjected to a few tenths of a second of extra delay. This delay may even be avoided, by forwarding the user to the IDP immediately after the IDP selection (47), so that the user can start performing authentication while the hub and RP continue the protocol. Then, the hub would "catch up" with the IDP (56) via a direct connection—this would only require an extra auxiliary random session ID decided unilaterally by the hub.

**Audit action.** The hub can prove to an auditor that the session ID in the signed request $\{n'\}$ sent to IDP (56) is the only one possible based on the signed request $\{n\}$ received from RP (53). For that, the hub opens the triplet (RP, IDP, $n_{\mathrm{H}}$) from $\chi_{\mathrm{H}}$ (49) and then the auditor verifies that $\text{CR-Hash}(\mathrm{RP}, n_{\mathrm{R}}, \chi_{\mathrm{H}}) = n$ and $\text{CR-Hash}(\mathrm{IDP}, r) = n'$, with $r = \text{CR-Hash}(n, n_{\mathrm{H}})$, where "RP" and "IDP" are the identifiers of the RP and IDP that signed the requests with IDs $n$ and $n'$, respectively.