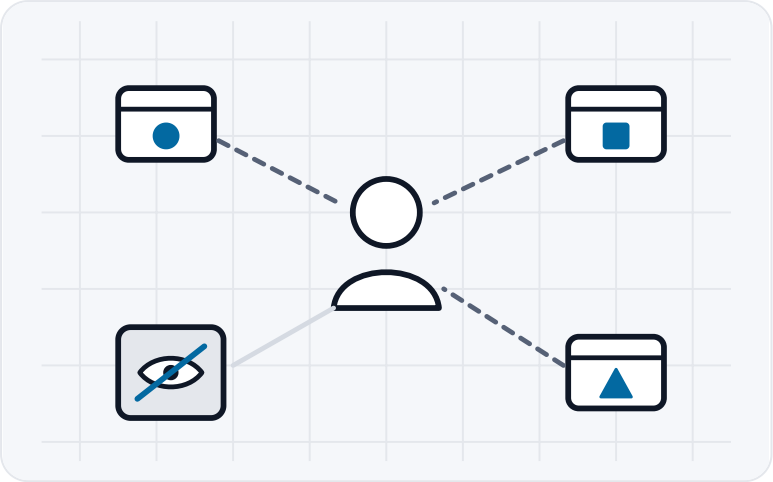
A-SIT Technology

E-Government Innovation Center

4 September 2024 Overview · Privacy & Crypto · Digital Identity · 18 min read

Privacy-preserving Identifiers

Stefan More



Authors Stefan More

Source https://technology2.a-sit.at/en/publications/privacy-preserving-identifiers/

Key findings

- Global identifiers allow service providers and identity providers to link a person's activities across platforms.
- Conventional PPID derivation and unobservability towards the identity provider are mutually exclusive.
- OPRF-based schemes such as BISON achieve unlinkability and unobservability without persistent state on the device.
- Achieving unobservability, unlinkability and Sybil resistance at the same time requires a protocol-aware browser.
- MPC-based identifiers could be more flexible than OPRF, but raise the question of who operates the nodes.

In federated authentication protocols (e.g., OpenID Connect), users are often assigned unique identifiers to be identified by service providers (SPs). This is, for example, used to re-authenticate a returning user at an online service. A resulting disadvantage is that the user's behavior can be linked across multiple SPs, which impacts the user's privacy (linkability). Alternatively, the identity provider (IDP) can derive a separate identifier for each SP (SP-scoped identifier). However, in current methods, this is always done directly by the IDP, which allows the IDP to observe at which SPs a user authenticates (observability).

The goal of a privacy-preserving identifier is thus to enable both unlinkability and unobservability. In a system using such identifiers, users are not traceable between different services, and identity providers cannot observe the user's behavior.

This project aims to analyze and discuss various methods for privacy-preserving SP-scoped user identifiers. Systems/methods based on cryptographic techniques such as OPRF, MPC, and ZKP will serve as examples.

1. Introduction

In today's digital world, privacy is a growing concern, especially when it comes to online identities. Federated authentication systems, such as OpenID Connect (OIDC), allow users to log into multiple services using a single account managed by an identity provider (IdP).

While this simplifies access for users, it also creates privacy challenges. One of the main issues is linkability, where a user's activity can be tracked across different service providers (SPs) using a unique identifier. Additionally, the IDP can observe which services a user accesses, leading to observability concerns.

Privacy-preserving identifiers aim to address these issues by providing solutions that enhance user privacy. Specifically, they seek to ensure that users cannot be tracked between different services (unlinkability), and that identity providers are unable to monitor user activity (unobservability). Through the use of advanced cryptographic techniques such as Oblivious Pseudo-Random Functions (OPRF), Multi-Party Computation (MPC), and Zero-Knowledge Proofs (ZKP), it is possible to create systems where users can authenticate securely while maintaining their privacy.

For example, consider a user logging into multiple online services such as a social media platform, an online store, and a news website, all through the same IdP. In a typical federated authentication system, the same unique identifier is used across all services, allowing both the service providers and the IdP to track the user's behavior and link their activity across these different platforms. This creates a detailed profile of the user, compromising their privacy.

In contrast, with a privacy-preserving identifier system, each service provider would receive a unique, SPscoped identifier for the same user. This ensures that neither the service providers nor the IdP can link the user's activities across platforms, significantly enhancing privacy. For example, when the user logs into the social media platform, the online store, and the news website, each interaction remains isolated, preventing any entity from building a complete picture of their behavior online.

1.1. Example

Example federated authentication („Social Login") options at a service provider.

Imagine a user, Alice, who frequently logs into various websites using her Google account, a common identity provider (IdP) in federated authentication systems. She uses her Google login to access a streaming service, an online shopping platform, and a fitness app. In the traditional federated model, each of these services—let's call them StreamNow, ShopEasy, and FitTrack—receives the same unique identifier for Alice from Google, e.g., her GMail address.

This identifier allows StreamNow, ShopEasy, and FitTrack to recognize Alice each time she logs in. However, because the same identifier is shared across all three platforms, it becomes possible for these services to combine information about Alice's behavior, like the shows she watches, the products she buys, and her fitness routines. Even worse, Google, the IdP, can also see when and where Alice logs in, giving it a complete picture of her online activities.

Now, in a privacy-preserving system, when Alice logs into StreamNow, ShopEasy, and FitTrack, Google would generate a unique identifier for each service. StreamNow would receive one identifier, ShopEasy another, and FitTrack a third. This ensures that Alice's activity on these platforms remains siloed—no service can link her behavior across platforms. If additional privacy is provided, Google cannot track which services she is using; however, this is not the case for all privacy-preserving identifier approaches (see below). If fully applied, such systems provide Alice with a much higher level of privacy.



More generally, pseudonyms serve as a privacy-enhancing mechanism by allowing users to interact with Service Providers without revealing their true identities or a persistent identifier.

One common approach to pseudonymity is Pairwise Pseudonymous Identifiers. Pairwise Pseudonymous Identifiers involve the creation of unique identifiers for each pairwise relationship between a user and a Service Provider. This means that each user generates a new pseudonym for each Service Provider it interacts with, and does not re-use the pseudonym from one SP with another SP. These identifiers are used as temporary or one-time pseudonyms during interactions, preventing SPs from tracking users across different services or sessions (Linkability).

2. Federated Authentication on the Web

The process of authentication ("Login") on the web typically involves the following actors [1]:

User The user is typically a person that wants to login at some service. More generally, the user is a party with the primary objective of proving their identity to some service securely and efficiently, ideally in a privacypreserving manner. In the web, the user is using a web browser to access various services.

Service Provider (SP) The service provider is a party that operates a service. It requires the user to authenticate, as part of an authorization decision to access some service or resource. There are many different service providers, which may offer wildly disparate services. An internet message board, a virtual storefront, a health provider's appointments software, a bank's online banking application, and a government bureau's web portal, are very different kinds of service providers, with different needs.

Almost all service providers share a need to (re-)identify users. If a user visits a message board, they should have the ability to edit messages they have previously sent; if they visit a storefront, they should be able to see their pending orders; if they visit their health provider, they should be able to see and cancel their appointments; and so forth. It is thus necessary to perform an authentication process of some kind, which results in a trusted persistent identifier for the user.

Depending on the method, the server might need to securely store persistent information – passwords, shared secrets, or recovery codes – for each user. Conversely, users will then also need to remember many different credentials for different service providers.

Since this is a burdensome task, this has led to a desire to delegate the actual authentication process to a dedicated third party. The resulting model is called federated authentication.

Identity Provider (IdP) The identity provider is a party that is trusted to perform user authentication. It decides on the authentication factors to use, and stores any information necessary to (re-)authenticate the user using these factors, such as passwords and various second factors or passkeys.

The identity provider then assigns each user account a "global" identifier, which is only unique within the context of this particular IdP. Some identity providers may use truly global identifiers, such as email addresses; but not all do.

Depending on the context, possession of an account at a certain identity provider may carry particular implications, such as membership in a certain organization, which is verified by the identity provider. In such a case, the service provider delegates not only authentication, but implicitly also verification of these properties, to the identity provider.

This approach allows service providers to securely re-authenticate users without the need to handle the actual authentication (and identification) process. This, it is quite common on today's web, e.g., in the form of social logins ("Login with Google", "Login with Facebook", or some E-ID systems like ID Austria).

Basic authentication flow in federated authentication.

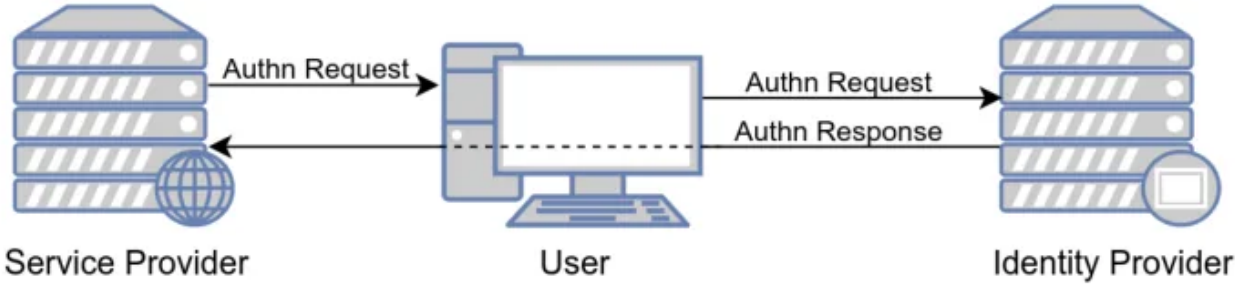
2.1. Federated Credential Management (FedCM)

Traditional OpenID Connect authentication flows, as described above, assume, and work around, a protocol-unaware user device. This is achieved by leveraging general-purpose technologies, such as HTTP redirects and HTML form submissions, to trigger the user's browser into forwarding opaque information between logically unrelated remote servers.

Recently, those technologies have come under scrutiny. This step comes in the wake of web browsers limiting the traditional ways of tracking user behavior across disparate web origins, such as third-party cookies, which have long been misused by the online advertising sector. In reaction, privacy adversaries

have explored new ways of associating user interactions; this includes the use of message-carrying (bounce) redirects to make the browser identify itself to an advertising tracker. In response, Browser manufacturers propose limiting such stateful redirects. While improving the user's privacy in the context of tracking, this limitation would eliminate the technology that OpenID Connect, and similar use cases, depend on.

The current W3C proposal to solve this conundrum is the Federated Credential Management (FedCM) API [7]. FedCM envisions browsers being conceptually aware of authentication processes, and taking an active role in negotiating user authentication. This represents a paradigm shift away from the traditional OIDC-unaware user device. An additional advantage of OIDC-aware browsers is that the browser can now play an active role in the authentication, instead of merely passing around data between SP and IdP.



This advantage is used by several approaches to enhance the user's privacy during federated authentication. [1]

3. Functional Requirements & Privacy Goals

3.1. Functional Requirements

The following requirements are the basis for a secure authentication system [1]:

- Soundness: If a user successfully completes an authentication process, this should guarantee that they have successfully authenticated to the identity provider as the user corresponding to the resulting pseudonym. In other words, a malicious user should not be able to – without the identity provider's cooperation – impersonate benign users towards a service provider.
- Validity: It should be possible for the identity provider to suspend or delete a user account, disabling that user's ability to authenticate.
- No client-side state: While storing secret information on the user's device can be very convenient for protocol design, it presents many practical challenges. Devices may break or be misplaced, leading to the irrevocable loss of any cryptographic information stored. Backing up such information while keeping it truly private is an open challenge. Users may also wish to log in on a shared computer, or on a friend's device. The process of transferring cryptographic key material to such a device is often complex, and exposing key material to a potentially-untrusted device is undesirable. Thus, we focus on approaches that don't require persistent state on the user device. All state kept on the user

device should be ephemeral to a single authentication process, and should be able to be discarded after the process completes. This matches existing protocols deployed in the real world, and avoids complications when integrating our scheme.

3.2. Privacy Goals:

In the context of (federated) authentication on the web, the following privacy goals are relevant [3,6]:

- Unlinkability: Users expect that they can use a credential or a set of credentials with different service providers without reducing their privacy. Specifically, the involved service providers should not be able to link the interactions with each other, which would allow for user profiling. In the case of anonymous/pseudonymous interactions, users would like to use the same credential twice (i.e., for age verification) in two interactions. The expectation is that the service cannot link the two interactions with each other, i.e., does not know that the same user interacted with it twice (multishow unlinkability).
- Unobservability: In the same way, users expect that no party except those directly involved in an interaction (i.e., the user and the service provider) learn that the interaction takes place. Specifically, users expect that the issuer or some other authority (e.g., the government) does not learn at what SP the user is using their wallet, or even that the user is using their wallet.[1]

4. Approaches

4.1. Non-pseudonymous Identifiers

The baseline for federated authentication on the web is an identifier without any added privacy measures: The user logs in at the identity provider (IdP), and the IdP returns a static identifier to the service provider (SP), i.e., the user's email address. This identifier is the same regardless which SP the user wants to access, and hence we call this identifier a global identifiers.

A global identifier is a unique identifier assigned to a user by an identity provider that remains consistent across all service providers. This type of identifier is typically used in federated authentication systems where users log into multiple services using the same credentials managed by a single IdP. The global identifier allows SPs to recognize the user across different platforms, providing a seamless experience where the user's identity is uniformly acknowledged no matter which service they access.

However, while convenient, global identifiers pose significant privacy risks. Since the same identifier is used across multiple services, it becomes possible for SPs and other entities to link a user's activities across platforms, learning about their behavior and compromising their privacy.

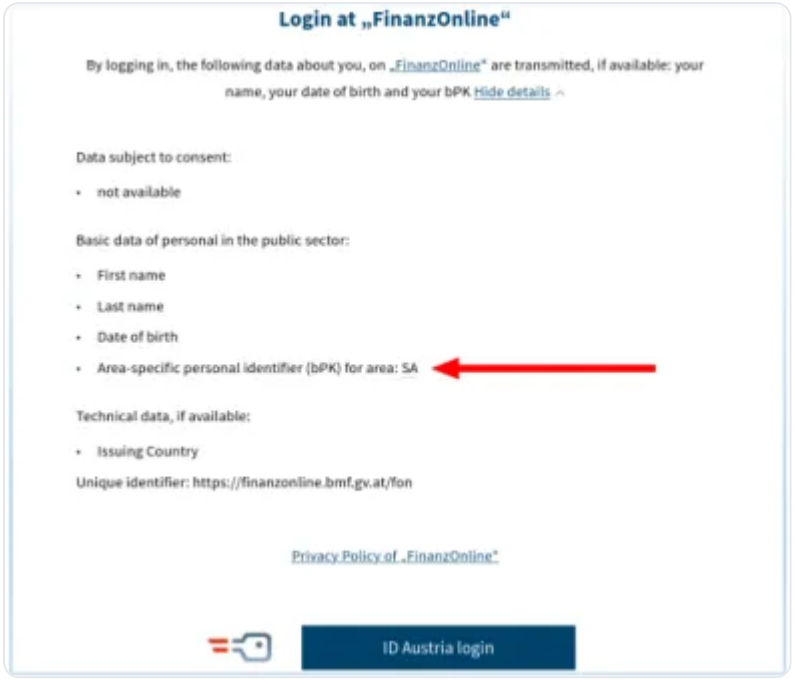
4.2. Pairwise Pseudonymous Identifier (PPID)

To provide better privacy to users, the OpenID Connect specification [8] allows for the use of pairwise pseudonymous identifiers (PPIDs) [1]. Here, the identity provider derives a pseudonymous identifier for a particular audience and substitutes it for the user's global identifier. For example, Apple's "Sign in with Apple" OpenID provider generates a different pseudonymous identifier based on the service provider's associated Apple Developer Account [9].

The specific derivation method, and audience delineation, are left up to the identity provider. Commonly, a hash digest of the user's global identifier concatenated with some audience identifier is calculated and used as the user's PPID [8, Section 8.1]. Traditional PPID derivation envisions the IdP knowing which audience that the user is authenticating to. It enshrines the need for the IdP to learn sensitive association data with each login. Traditional PPID derivation and unobservability towards the identity provider cannot coexist.

4.3. Sector-specific Personal Identifier (ssPIN, bPK)

Another form of privacy-enhanced identifiers are Sector-specific Personal Identifiers (ssPINs), also known as a Bereichsspezifisches Personenkennzeichen (bPK) in Austria, is a privacy-enhancing method used to generate unique user identifiers for different sectors or service providers.



Example login with an Austrian E-ID, showing that instead of a global identifier only the sector-specific identifier ("bPK") for the "Taxes and Expenses (SA)" sector is shared with the Treasury SP.

The idea behind ssPIN is to provide each sector or service provider with a separate identifier for the same user, ensuring that activities across different sectors cannot be linked. For example, a user interacting with healthcare services would have one identifier, while using banking services would generate a completely different identifier, making cross-sector tracking impossible. These identifiers are typically derived from a user's primary identifier (such as a national ID) using cryptographic methods, ensuring that the user's identity remains consistent within a sector while being unlinkable across sectors. By isolating user identities in this way, ssPIN enhances both unlinkability, preserving privacy across different domains of service.

4.4. OPRF-based Blind Identifiers (BISON, OPPID)

Another proposal to provide privacy-preserving authentication in federated systems is the BISON protocol [1]. BISON stands for Blind Identification with Stateless scOped pseudoNyms and addresses some of the major privacy risks associated with traditional systems like OpenID Connect.

The key feature of BISON is that it enables pseudonym derivation in a way that is both unlinkable and unobservable. This means that it prevents identity providers (IdPs) from tracking which service providers (SPs) a user interacts with, while also ensuring that service providers cannot link user activity across different services. BISON achieves this through a cryptographic technique called Oblivious Pseudorandom Functions (OPRF), which allow the IdP to help generate pseudonyms without knowing the exact SP for which the pseudonym is being generated. This is done with the browsers help (cf. Section 4.5 below) and thereby offers a much stronger level of privacy than traditional pairwise pseudonymous identifiers, where the IdP can still observe user interactions.

Another important aspect of BISON is that it does not require storing long-term state on the user's device, making it lightweight and easy to integrate with existing system. Additionally, BISON introduces a small extension that allows PPID pseudonyms to be generated using the BISON protocol. This makes it a practical solution for real-world privacy concerns, especially as it provides strong privacy guarantees without adding complexity to the authentication process.

Two other proposals similar to BISON are OPPID (Oblivious Pairwise Pseudonyms) [2] and UPPRESSO (Untraceable and Unlinkable Privacy-PREserving Single Sign-On Services) [10]. Using zero-knowledge proofs (ZKPs), OPPID also achieves SP Authentication, i.e., allowing the system to limit which entities can function as SP.

4.5. Excursus: Client-side Modifications

One crucial benefit of OpenID Connect, which has contributed to its widespread adoption, is that it does not require the user device to be aware of its existence. Client-augmented OIDC does not offer this benefit. It requires the web browser to take an active role: verify the SP's identifier (scope), perform blinding operations, and obtain user consent using built-in dialog windows. This requires the web browser to be aware of OIDC's existence.

However, a protocol-aware browser is needed to achieve unobservability, unlinkability, and sybil resistance simultaneously [1].

To show this, consider authentication as a two-party protocol between a service provider and an identity provider using a protocol-unaware browser as a communication channel. Assume that we have two distinct service providers, A and B, and a user, X.

The identity provider now receives an authentication request. At this point, the request must not allow the identity provider to determine whether it originated from A or B; if this were possible, we would lose unobservability. After some user interaction to authenticate X, some data is returned to the originator, which results in identifier P. Since we assume Sybil resistance, P needs to be stable across authentication requests. However, recall that the request might have originated from either A or B; therefore, it cannot provide unlinkability.

4.6. MPC-based ZKP-based Identifiers

A potential research direction are MPC-based identifiers. In MPC (Multi-party computation), a group of nodes (at least two) collaboratively compute a function. This can be used to achieve protocols in which multiple parties input data into the function while ensuring that no party learns the input of the other parties. In the end, one or all nodes learn the computation's result.

This can be used to construct an identifier derivation protocol similar to the one described in the BISON section above: First, the user authenticates at the IdP. It then initiates a login at some SP and inputs the SP's scope identifier in the MPC computation. On the other end, the IdP inputs the user's account identifier into the computation. As a result, the user learns the SP-specific identifier (the login is unlinkable). Meanwhile, the user does not learn their global identifier and, most importantly, the IdP does not learn the SP's identifier (the request remains unobservable). A challenge in the context of MPC protocols is the question of who is maintaining and operating the nodes (incentive structure, trust setup, etc.). A potential advantage of MPC over OPRF-based protocols could be that MPC protocols are more flexible, and can perform other checks in addition to the identifier computation (e.g., access policy checks [11]).

Another special form of privacy-preserving identifier are cryptographic pseudonyms based on zeroknowledge proofs (ZKP), e.g., ABC4Trust Pseudonyms [5]. They provide anonymous (or pseudonymous) and unlinkable usage of a service but enable a trusted entity (Inspector) or set of entities to later reveal the identity of a pseudonym.

By using cryptographic (key) material stored on the user's device (e.g., phone, smartcard or browser), they enable users to locally derive their own pseudonyms. By doing so, the IdP is not directly involved in the process of authenticating and can thus not observe the user's behavior. Additionally, since a user can generate a new pseudonym for each service provider, there is no linkability between the different authentication processes at different service providers. However, this approach requires local state in the user domain and is thus not suited for all use cases, especially in the context of web authentication.

5. Conclusions

In this report, we explored various approaches to implementing privacy-preserving identifiers in federated authentication systems, focusing on unlinkability and unobservability. Traditional federated systems, which use global identifiers, pose significant privacy risks, as they allow both service providers (SPs) and identity providers (IdPs) to link a user's activities across platforms. To address these concerns, several advanced cryptographic techniques have been proposed and discussed.

As the demand for privacy-preserving authentication grows, it is likely that these techniques will see broader adoption, particularly as web standards evolve to accommodate more privacy-centric models. Future work should explore the trade-offs between privacy, usability, and performance, as well as the potential for realworld implementation in diverse sectors, from social media to e-government services.

References

[*] OpenAI. ChatGPT (4o) [Large language model] was used to edit parts of the text.

[1] Jakob Heher, Stefan More, Lena Heimberger, BISON: Blind Identification with Stateless scOped pseudoNyms. arXiv pre-print, 2024.

[2] Maximilian Kroschewski, Anja Lehmann, Cavit Özbay, OPPID: Single Sign-On with Oblivious Pairwise Pseudonyms. Cryptology ePrint Archive, 2024.

[3] Stefan More, Jakob Heher, Edona Fasllija, Maximilian Mathie, Service Provider Accreditation: Enabling and Enforcing Privacy-by-Design in Credential-based Authentication Systems. Proceedings of the 19th International Conference on Availability, Reliability and Security, 2024.

[4] Stefan More, Trust and Privacy in a Heterogeneous World. PhD Thesis, Graz University of Technology, 2023.

[5] K. Rannenberg, J. Camenisch, A. Sabouri, Attribute-based credentials for trust. Springer Cham, 2015.

[6] A. Pfitzmann, M. Hansen. Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology (Version 0.28), 2006.

[7] Nicolás Peña Moreno, Sam Goto, Federated Credential Management API (Working Draft). https://www.w3.org/TR/fedcm, August 2024.

[8] Nat Sakimura, John Bradley, Michael B. Jones, Breno de Medeiros, Chuck Mortimore, OpenID Connect Core 1.0. https://openid.net/specs/openid-connect-core-1_0.html, 2014.

[9] Apple, Authenticating users with “Sign in with Apple”. https://developer.apple.com/documentation/sign_in_with_apple/sign_in_with_apple_rest_api/authenticating_users_with_sign_in_with_apple, retrieved Sept 2023.

[10] Chengqian Guo et al., UPPRESSO: Untraceable and Unlinkable Privacy-PREserving Single Sign-On Services. arXiv pre-print, 2022.

[11] Stefan More, Lukas Alber, YOU SHALL NOT COMPUTE on my Data: Access Policies for Privacy Preserving Data Marketplaces and an Implementation for a Distributed Market using MPC. Proceedings of the 17th International Conference on Availability, Reliability and Security, 2022.

---

1. We argue in Section 4.5 that protocol-aware clients are not only beneficial, but necessary, to provide both unlinkability and unobservability at the same time. ↩