Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.066139

REVIEW

# Single Sign-On Security and Privacy: A Systematic Literature Review

**Abdelhadi Zineddine[1,#], Yousra Belfaik[2,#], Abdeslam Rehaimi[1], Yassine Sadqi[3,*] and Said Safi[1]**

[1]Laboratory LIMATI, FPBM, Sultan Moulay Slimane University, Beni Mellal, 23000, Morocco
[2]Laboratory ISIMA, FPT, Ibn Zohr University (UIZ), Taroudant, 83000, Morocco
[3]Laboratory L2IS, FST, Cadi Ayyad University, Marrakech, 40000, Morocco

*Corresponding Author: Yassine Sadqi. Email: yassine.sadqi@ieee.org

[#]These authors contributed equally to this work

Received: 31 March 2025; Accepted: 18 June 2025; Published: 30 July 2025

**ABSTRACT:** With the proliferation of online services and applications, adopting Single Sign-On (SSO) mechanisms has become increasingly prevalent. SSO enables users to authenticate once and gain access to multiple services, eliminating the need to provide their credentials repeatedly. However, this convenience raises concerns about user security and privacy. The increasing reliance on SSO and its potential risks make it imperative to comprehensively review the various SSO security and privacy threats, identify gaps in existing systems, and explore effective mitigation solutions. This need motivated the first systematic literature review (SLR) of SSO security and privacy, conducted in this paper. The SLR is performed based on rigorous structured research methodology with specific inclusion/exclusion criteria and focuses specifically on the Web environment. Furthermore, it encompasses a meticulous examination and thematic synthesis of 88 relevant publications selected out of 2315 journal articles and conference/proceeding papers published between 2017 and 2024 from reputable academic databases. The SLR highlights critical security and privacy threats relating to SSO systems, reveals significant gaps in existing countermeasures, and emphasizes the need for more comprehensive protection mechanisms. The findings of this SLR will serve as an invaluable resource for scientists and developers interested in enhancing the security and privacy preservation of SSO and designing more efficient and robust SSO systems, thus contributing to the development of the authentication technologies field.

**KEYWORDS:** Single sign-on; authentication; OAuth2.0; OpenID connect; security; privacy; mitigation solutions

## 1 Introduction

In today's digital landscape, it has become increasingly common for individuals to have multiple online accounts across various platforms. In February 2025, the number of social media platforms' users was approximately 5.24 billion out of 5.56 billion internet users worldwide, and that number is rising each year [1]. Managing multiple accounts with different usernames and passwords, from social media to email, banking, and shopping websites, can be challenging. Due to human memory limitations, users struggle to remember passwords for numerous accounts. Consequently, this leads to using weak and easily guessable passwords or adopting insecure account management techniques such as reusing passwords [2–4]. Single Sign-On (SSO) solutions have emerged as a promising approach to alleviate this burden, allowing users to seamlessly access multiple services using a single set of credentials. Due to its superior usability to reduce password fatigue and ensure that users have strong authentication and account monitoring measures in place when accessing third-party services, SSO has seen extensive adoption over the past decades [5]. The SSO mechanism allows a user to seamlessly access multiple services or applications called Relying Parties (RPs) or Service Providers

(SPs) using their identity on a trusted entity charged to authenticate users and store their login credentials securely called Identity Provider (IdP) [6]. This process enables users to access various RPs by obtaining a token containing their authentication credentials from the IdP, thus eliminating the need to remember multiple passwords.

Numerous protocols have been developed for implementing SSO-based systems. Among these, OAuth 2.0 and OpenID Connect have become the most widely adopted and are currently regarded as industry standards for enabling secure and scalable authorization and authentication in modern web applications [7,8]. Their architecture, which relies on lightweight JSON structures and RESTful APIs, facilitates seamless integration, particularly with JavaScript-based frontends. This has contributed significantly to their widespread adoption. These protocols are employed by several prominent identity providers to deliver SSO functionality, including platforms such as Google, Facebook, Microsoft, Amazon, and GitHub [9,10]. In contrast, while protocols like SAML continue to be applicable in traditional enterprise environments, their reliance on XML and the associated implementation complexity can pose limitations in dynamic or large-scale web ecosystems [11].

OAuth 2.0 is an authorization framework that enables users to provide restricted access to their resources on an application to another (i.e., web app, native app, or API service) without sharing their credentials. The OAuth 2.0 specification was introduced by the Internet Engineering Task Force (IETF) in October 2012 as *RFC 6749: The OAuth 2.0 Authorization Framework* [12]. It involves the interaction of four roles: resource owner, client, resource server, and authorization server. The resource owner refers to the entity that possesses the authority to authorize access to a protected resource (i.e., the end-user), while the client is the application requesting access to the protected resource (the client is itself the RP when OAuth2.0 is used for SSO). The resource server is responsible for hosting the protected resources and validating the access tokens given by the client. On the other hand, the authorization server is responsible for verifying the identity of the resource owner and issuing access tokens to the client [13]. When using OAuth 2.0 for SSO, the resource server and the authorization server jointly constitute the IdP. These roles work together to ensure secure authorization and access control in OAuth 2.0. It is crucial to understand that OAuth 2.0 is mainly an authorization protocol that grants access to protected resources and does not include an authentication layer. Authentication is the procedure of confirming the identity of a user, while authorization determines what actions a user is allowed to perform. OpenID Connect (OIDC) is a protocol that enables delegated authentication by establishing an identity layer on top of the OAuth 2.0 protocol [14].

OIDC was developed in November 2014 by the OpenID Foundation. OIDC addresses the need for authentication in OAuth 2.0 by providing a standardized way to verify the identity of users [15]. It allows applications to obtain user information, such as name and email address, from an IdP and use it for authentication purposes. OIDC enables RPs to confirm end-user identity by introducing a new type of token to OAuth 2.0, called the ID token, in addition to the existing access token and authorization code [10]. The ID token includes the user's identification information, which is digitally signed by the IdP to ensure its authenticity. RPs can use this ID token to securely authenticate and authorize users, providing a seamless and secure user experience. In the Web environment, the SSO system has experienced significant growth. In a study conducted by Ghasemisharif et al. [16], the authors analyzed a compilation of the top 1 million websites based on Alexa rankings. Their findings revealed that 57,555 websites (6.30% of the list) support SSO. In 2020, Zhang et al. [17] reported that over one million websites now support SSO through the OpenID Connect protocol. This indicates a substantial increase in the adoption of SSO among web platforms. The Table 1 lists the key acronyms used later in this study.

**Table 1:** List of key acronyms

| Acronym | Definition | Acronym | Definition |
|---|---|---|---|
| APIs | Application Programming Interfaces | LR | Literature Review |
| CAs | Certification Authorities | MITM | Man-in-the-Middle |
| CP | Conference Proceeding | MPN | Microsoft Partner Network |
| CSRF | Cross-Site Request Forgery | OAuth | Open Authorization |
| DDoS | Distributed Denial of Service | OIDC | OpenID Connect |
| DID | Decentralized Identifiers | PII | Personally Identifiable Information |
| eID | Electronic Identification | RPs | Relying Parties |
| EC | Exclusion Criteria | RQs | Research Questions |
| FIDO | Fast Identity Online | SAML | Security Assertion Markup Language |
| HTTPS | HyperText Transfer Protocol Secure | SLM | Systematic Literature Mapping |
| IAM | Identity and Access Management | SLR | Systematic Literature Review |
| IC | Inclusion Criteria | SPs | Service Providers |
| IdM | Identity Management | SSI | Self-Sovereign Identity |
| IdP | Identity Provider | SSO | Single Sign-On |
| IETF | Internet Engineering Task Force | TPM | Trusted Platform Models |
| IPFS | InterPlanetary File System | WRA | Wildcard Receiver Attack |
| JA | Journal Article | XSS | Cross-Site Scripting |
| JWT | JSON Web Token | | |

### *1.1 Paper Motivation*

Despite the benefits of SSO, it raises various security and privacy concerns that need to be addressed. For example, in October 2016, Yahoo claimed that over 1 billion accounts had been affected by a security breach. The central user database used for user authentication across all Yahoo products like Yahoo Mail, Sports or Finance, was compromised. Therefore, the attackers not only obtained the usernames and email addresses of the affected users, but also additional confidential information, including dates of birth, phone numbers, hashed passwords, and unencrypted security responses [18]. In April 2018, Facebook alerted its 2.2 billion users that their public profile information may have been compromised by malicious third-party scrapers. Mark Zuckerberg, Facebook's CEO, stated that these malicious actors used Facebook's "Search" functions to gather information on a large portion of its global user base. This revelation highlights Facebook's ongoing failure to safeguard user privacy, despite the significant revenue it makes from the same data. This announcement follows the Cambridge Analytica scandal, in which the political consulting business inappropriately accessed and abused personal data from 77 million users, reportedly affecting the 2016 US presidential election [19]. In February 2023, Microsoft took action to remove fraudulent MPN accounts that were being utilized to create malicious OAuth apps as a part of a phishing scam. The objective of this campaign was to compromise organizations' cloud environments and pilfer email data. The fraudulent actors behind this scheme utilized the malicious apps to conduct a consent phishing campaign, tricking users into granting permissions to these fraudulent applications. Then, these applications can be exploited to get unauthorized access to legitimate cloud services and sensitive user data [20]. This demonstrates that an insecure and ineffective SSO system can have potentially disastrous implications, such as unauthorized access to sensitive information and resources, data breaches, identity theft, disruption of the system's availability and functionality, users' privacy leakage, etc.

Therefore, we chose to conduct a systematic literature review in this paper on the SSO ecosystem's security and privacy in the web environment. This study aims to identify the existing gaps in current SSO systems, explore potential mitigation mechanisms and solutions that can be used to address the identified threats, understand the current state of the art of SSO systems, and pinpoint the specific security and privacy weaknesses that require improvement or further research. Through a comprehensive analysis of the existing literature, we aim to consolidate a coherent body of knowledge on the topic. This knowledge will be valuable to assist researchers, practitioners, and policymakers in better understanding the weaknesses in their SSO-based systems, developing best practices and defense mechanisms, and taking proactive measures and strategies to ensure the long-term security and privacy and the overall trustworthiness of SSO systems.

### *1.2 Paper Contributions*

To the best of our knowledge, this SLR is the first of its kind in the literature since no previous work has provided a large-scale review on SSO implementations and its standard protocols, OAuth 2.0 and OpenID Connect, while simultaneously addressing both security and privacy aspects. The main contributions of this review are as follows:

- Identify the various security and privacy threats associated with SSO implementations and its standard protocols, OAuth 2.0 and OpenID Connect, as well as pinpoint their implications and the attacks that can exploit each of the identified threats.
- Explore and analyse the different mitigation techniques and measures proposed in the literature that can be used to address the identified SSO security and privacy threats.
- Understand the state of the art of the current solutions and schemes proposed in the literature addressing SSO security and privacy weaknesses to assess the effectiveness of the different mitigation techniques.
- Categorize and classify the different mitigation techniques based on the targeted threat, assessing organizations and practitioners to choose the most effective technique to their needs and save significant investments and resources.
- Highlight the specific weaknesses where further research or improvement is needed, enabling researchers and practitioners to focus their efforts on addressing these critical areas for advancement in the field.

### *1.3 Paper Outline*

The remainder of this paper is organized as follows: Section 2 presents the related works. Section 3 describes the research methodology followed in this study. Section 4 provides the findings of the conducted SLR, answering all the highlighted research questions. Finally, Section 5 concludes the paper and specifies future works. Fig. 1 illustrates the overall outline of this systematic literature review.

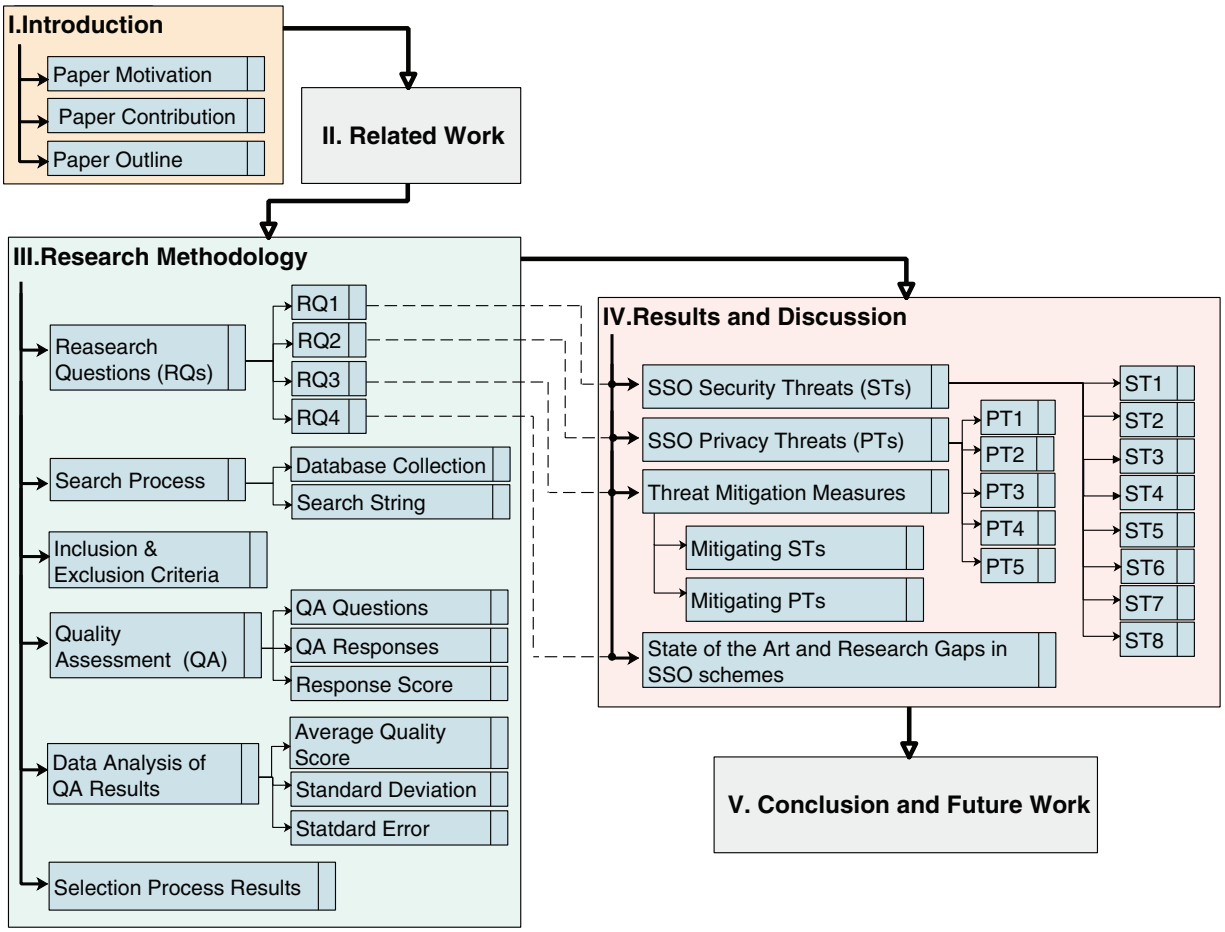


**Figure 1:** The overall outline of this SLR

## 2 Related Work

This section provides an overview of studies that have undertaken a literature review on the SSO system or any other similar system. Table 2 presents a list of surveys, literature reviews or systematic literature reviews that have addressed SSO, IAM, IDM, or any other authentication and authorization system, while specifying the environmental focus, data sources, publication types and search methodology used in each study.

**Table 2:** State of the art concerning SSO reviews

| Ref. | Year | Study type | Addressed system | Security/ Privacy | Environment | Data sources | Data type | Search methodology |
|---|---|---|---|---|---|---|---|---|
| [21] | 2012 | Survey | SSO | Security | – | – | – | – |
| [22] | 2016 | LR | SSO | Security | Cloud | – | – | |
| [23] | 2018 | Survey | SSO | Security | – | – | – | – |
| [24] | 2022 | SLM | IdM | Security and privacy | Web | IEEE Xplore, ACM, Science Direct, Springer and Wiley. | – | Kitchenham et al. (2011) |

(Continued)

**Table 2 (continued)**

| Ref. | Year | Study type | Addressed system | Security/ Privacy | Environment | Data sources | Data type | Search methodology |
|---|---|---|---|---|---|---|---|---|
| [25] | 2022 | SLR | Authentication and authorization in microservices | Security | – | DBLP, IEEE Xplore and Scopus. | – | Kitchenham et al. (2007) |
| [26] | 2023 | SLR | APIs | Security | Web and mobile | Scopus, IEEE Xplore, and ACM. | CP and JA | Kitchenham et al. (2007) |
| [27] | 2023 | SLR | IAM | Security | Enterprises | ACM, AISeL, EBSCO, EconBiz, Emerland Insight, Google scholar, IEEE Xplore, JSTOR, Springer, and Web of science. | CP and JA | Levy and Ellis (2006) and Kitchenham et al. (2009) |
| [28] | 2023 | LR | IdM | Security | Web and mobile | – | – | – |
| Our Work | 2025 | SLR | SSO | Security and privacy | Web | Scopus, Web of Science, ACM, IEEE Xplore, and Science Direct. | CP and JA | Kitchenham et al. (2009) |

Note: The symbol (–) indicates that the document analyzed is missing the criterion specified in the corresponding column.

In 2012, Radha et al. [21] performed a survey on the SSO system, its various types, how and where they are employed, and the types of credentials used in the complex SSO architecture. Moreover, it discusses the different protocols used on SSO, including Kerberos, SAML, OpenID, and BrowserID.

In 2016, Cusack and Ghazizadeh [22] conducted a brief literature review on single sign-on security risks in the cloud environment. The authors present an overview of the different identity management technologies and solutions used to manage users' access to cloud services. Additionally, they highlight the security challenges associated with SSO in the cloud, including identity theft, single point of failure, and user identity misuse. Next, the paper proposes a comprehensive solution to address these risks. This solution involves using OpenID's protocol along with Trusted Platform Models (TPM) for verification checks in order to meet user requirements while strengthening the trustworthiness and security infrastructure within SSO.

In 2018, Nongbri et al. [23] provided a survey on single sign-on, describing how it works and its different types. Subsequently, the authors describe the various protocols used in SSO, including OpenID, SAML, BrowserID, and Kerberos, along with their associated security concerns. Finally, the manuscript presents the advantages and drawbacks of token-based and PKI-based protocols.

In 2021, Rathee and Singh [24] conducted a systematic literature mapping on secure identity management (IdM) using blockchain technology. The authors provide an extensive review of the IdM system and its various challenges, including scalability, interoperability, and user privacy, with an emphasis on how the emergence of blockchain has addressed these challenges. The manuscript also aims to examine the various initiatives that have leveraged blockchain for IdM and identify the most widely used consensus mechanisms across different blockchain-based IdM frameworks.

In 2022, de Almeida et al. [25] performed a systematic literature review of authentication and authorization security concerns within microservices architectures. In this review, the authors aimed to identify the challenges and issues associated with authentication and authorization in microservices architecture. The authors also provide solutions that utilize open-source technologies to build security mechanisms. The authors also provide solutions that utilize open-source technologies for developing security mechanisms. Notably, their findings highlighted the effectiveness of mechanisms like OAuth 2.0, OpenID Connect, API Gateway, and JWT in mitigating the risk of unauthorized access to microservices if these mechanisms are implemented correctly.

In 2023, Mousavi et al. [26] conducted a Systematic Literature Review to provide a comprehensive understanding of the different types of security API misuses reported in the literature. To achieve this, the authors specifically focused on analyzing the usage patterns of six prominent security APIs: cryptographic primitives, SSL/TLS, OAuth, Fingerprint, Spring, and SafetyNet Attestation. Furthermore, the study examined the existing approaches developed for detecting security API misuses and evaluated their effectiveness. The authors also provided valuable recommendations for best practices based on their key findings and identified areas that require further research in the field of security API misuse detection. In the same year, Glöckler et al. [27] initially conducted a comprehensive review of existing literature to examine and comprehend the requirements for Identity and Access Management (IAM) systems from an enterprise standpoint. Additionally, they explored the potential advantages of self-sovereign identity (SSI), which is an emerging passwordless approach in identity management. SSI enables end-users to possess cryptographic attestations stored in digital wallet applications. Moreover, the authors categorized these IAM system requirements into four distinct aspects: security and compliance, operability, technology, and user considerations. In a subsequent phase, they proposed a prototype for IAM based on SSI that can fulfil requirements in all four categories of IAM system requirements. Twelve domain experts assessed this prototype's suitability for handling IAM issues. Another literature review was conducted in 2023 by Kiourtis et al. [28]. The authors provided a comprehensive overview of the current state of identity management (IDM) standards and regulations in information systems, including SAML, WS-Federation, OAuth, OpenID, FIDO, and Mobile Connect. These standards are essential for creating interoperable electronic identification (eID) systems guaranteeing data safety and secure access to sensitive information. The manuscript considers various architectural components, such as data domain and confidentiality needs, as well as different scenarios. Additionally, it explores the utilisation of short- and long-range distance data exchange protocols.

As illustrated in Table 2, various works provided research reviews to examine and understand the security concerns in IAM systems, IDM systems, authorization and authentication in microservices architecture, and security API misuses. Subsequently, only a few works have been conducted, either surveys or literature reviews on the single sign-on system, to understand its workflow and different protocols. However, no prior work has undertaken a large-scale systematic literature review on the SSO ecosystem and its standard protocols, OAuth 2.0 and OpenID Connect, in the web environment, while simultaneously combining both security and privacy.

## 3 Research Methodology

A literature review can be conducted using one of three methods: systematic review, semi-systematic review, or integrative review. In this study, we opted for the systematic literature review methodology to undertake a comprehensive investigation into the security and privacy aspects of single sign-on. The systematic review method was chosen due to its reputation for accuracy and rigour in literature reviews. It is recognized for its unbiased and thorough nature, as well as its transparent and replicable methodology, which helps minimize bias and ensure credibility [29]. To ensure a robust and standardized process, our

SLR follows the guidelines proposed by Kitchenham et al. [30]. The literature was filtered and selected in the following steps: define the research questions (1), describe the search process (2), specify the inclusion and exclusion criteria (3), apply quality assessment to the selected papers (4), and data synthesis (5).

### *3.1 Research Questions*

When embarking on an SLR, the first crucial step is to determine the research questions and the motivation behind answering them. To align with our study's goal, we have pinpointed four research questions (RQs) that explore the security and privacy issues related to SSO, their effects, and potential solutions. These questions serve as a foundation for identifying and selecting relevant studies for the review. Table 3 shows these research questions and explains the motivation for answering each one of them.

**Table 3:** Research questions and motivation

| Research question | Motivation |
|---|---|
| **RQ1:** What are the various security vulnerabilities and threats associated with SSO on the web environment, and what are their implications? | SSO is widely used for authentication in web-based applications. Identifying and understanding SSO security vulnerabilities and their implications is critical for researchers, developers, and security professionals to implement effective countermeasures and strategies. |
| **RQ2:** What are the privacy threats of SSO on the web environment, and what are their implications? | While SSO systems offer convenience and efficiency, it also raises privacy concerns. Therefore, it is mandatory to identify the numerous SSO privacy threats and understand their implications to help assess severity, identify research gaps, and propose innovative approaches to enhance privacy protection in SSO. |
| **RQ3:** What are the possible countermeasures that can be used to mitigate SSO security and privacy threats? | Identifying and analysing the mechanisms and measures that can be used to mitigate the various SSO security and privacy threats helps researchers and developers implement best practices, guidelines, and standards to enhance SSO security and privacy. |
| **RQ4:** What is the state of the art of the existing SSO systems, and what specific weaknesses still require further research? | Understanding the state of the art of the current SSO schemes proposed in the literature and identifying security and privacy areas that require improvement or further research. This helps make well-informed decisions when selecting and implementing SSO solutions to meet specific privacy and security requirements. |

### *3.2 Search Process*

To include relevant publications, we automated the search process using Web of Science[1] and Scopus[2] scientific databases, ACM[3] and IEEE Xplore[4] digital libraries and Science Direct[5] search engines. These sources offer an advanced search limiting the number of desired papers according to the year of publication,

[1] https://www.webofscience.com/wos/woscc/advanced-search (accessed on 17 June 2025)
[2] https://www.scopus.com (accessed on 17 June 2025)
[3] https://dl.acm.org (accessed on 17 June 2025)
[4] https://ieeexplore.ieee.org (accessed on 17 June 2025)
[5] https://www.sciencedirect.com (accessed on 17 June 2025)

article type and subject areas. For our SLR, we limit the search to the past eight years' publications (2017–2024). Furthermore, we only include publications of conference proceedings (CP) and journal articles (JA).

To answer the research questions of our study, we selected the most appropriate keywords and did some tests to determine the best combination and variation of keywords using the logical operators `AND` and `OR`. To find relevant papers related to the objective of this review, we have used the following keywords: `"Single sign-on"`, `"OAuth"`, `"OpenID Connect"`, `"security"`, `"privacy"`, `"attack"` and `"threat"`. We used an automatic search in the mentioned data sources above using the search string: `("Single sign-on" OR "OAuth" OR "OpenID Connect") AND ("security" OR "privacy" OR "attack" OR "threat")`. The initial search revealed 2315 publications after applying the time and document type filter.

### *3.3 Inclusion and Exclusion Criteria*

To ensure clarity and accuracy during the selection process, it is crucial to establish predefined criteria for inclusion and exclusion. These criteria help maintain rigour and objectivity throughout the process and ensure that only high-quality studies are included in the final review. Therefore, we have applied the inclusion (IC) and exclusion (EC) criteria outlined below to the publications retrieved in the preceding phase.

- **Inclusion criteria:**
  - **IC1:** Papers published between January 2017 and December 2024.
  - **IC2:** Papers published in journals and conferences, symposiums and workshops proceedings.
  - **IC3:** Papers that discuss SSO security and/or privacy vulnerabilities.
  - **IC4:** Papers that discuss security and/or privacy in OAuth and/or OpenID Connect protocols.
  - **IC5:** Papers that propose an SSO approach or scheme.
- **Exclusion criteria:**
  - **EC1:** Papers that are not written in English.
  - **EC2:** Duplicated studies in different databases (we keep only one copy).
  - **EC3:** The emphasis of the paper is not on the web environment.
  - **EC4:** The paper's full text is not accessible.
  - **EC5:** The paper's focus, after reading the title and abstract, is out of the aim of this review.

### *3.4 Quality Assessment*

After the inclusion and exclusion phase, the 130 selected publications will be evaluated according to the quality assessment (QA) questions. The papers' quality will be assessed by a list of questions to obtain a classification of the best-evaluated studies. Table 4 shows the quality assessment questions used in our review, along with the three possible responses "Yes", "Partly" and "No", and their corresponding scores: Yes = 2 points, Partly = 1 point, and No = 0 point. Each paper will be classified according to the total score achieved for the eight quality questions and the papers with a lower score than nine will be excluded. This ranking intends to highlight publications that provide more comprehensive information to address the research questions of our SLR. It is important to note that the exclusion of some articles does not imply that they are of worse quality compared to the included ones. Rather, it signifies that the top-rated papers offer a greater amount of information to address our specific questions. Based on the papers' quality analysis results, 88 papers were selected to answer the research questions of this SLR. Fig. 2 illustrates the quality assessment results.

**Table 4:** Quality assessment questions with corresponding potential scores

| $QA_i d$ | Questions | Response score | | |
|---|---|---|---|---|
| | | Yes (2 pt) | Partly (1 pt) | No (0 pt) |
| **QA1** | Does the study clearly explain the context and the scope? | – | – | – |
| **QA2** | Does the work define a vulnerability in SSO and its impact? | – | – | – |
| **QA3** | Does the work propose some defence mechanisms to mitigate a threat? | – | – | – |
| **QA4** | Does the work propose an approach or a solution for SSO? | – | – | – |
| **QA5** | Does the work conduct an empirical study or analysis of the proposed solution or discovered vulnerability? | – | – | – |
| **QA6** | Is the work compared with other similar works? | – | – | – |
| **QA7** | Does the work address both security and privacy? | – | – | – |
| **QA8** | Does the paper make significant contributions to the field of study? | – | – | – |

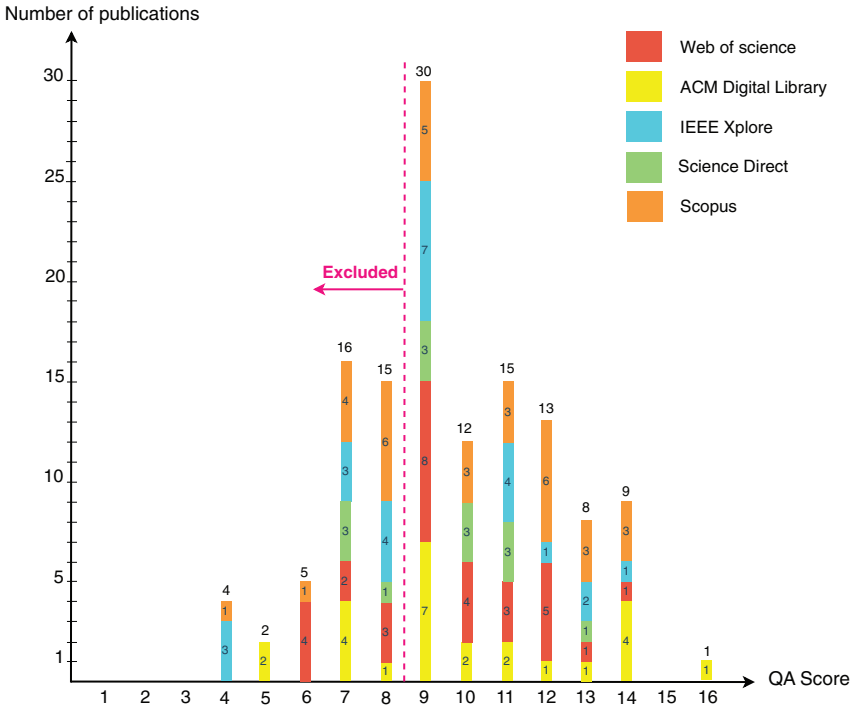


**Figure 2:** Quality assessment results

### 3.5 Data Analysis of QA Results

We examined the correlation between the quality score and the publication year of the paper. Table 5 presents the number of studies, average quality score, standard deviation, and standard error for each year from 2017 to 2024. The average quality score ranges from 9.64 in 2018 to a peak of 12.28 in 2022, followed by a slight decrease through 2024. The standard deviation, which provides information about how spread out the quality scores are from the average value, shows some fluctuation over the years. It ranges from 1.05 in 2018 to 2.05 in 2019 and 2.01 in 2022. Interestingly, the standard deviation is identical for the years 2021 and 2023, both recording a value of 1.44. Additionally, the standard error, representing the uncertainty of the average quality score, varies between 0.28 in 2018 and 0.54 in 2022, indicating the precision of the average estimates for each year. Notably, the standard error in 2022 is the highest, suggesting a slightly greater uncertainty around the quality score in that year. In summary, the average quality score demonstrates a consistent increase from 2018 to 2022, reaching its maximum before showing a slight decline from 2022 to 2024. Meanwhile, the standard deviation and standard error exhibit fluctuations, indicating changes in variability and the precision of the estimates across the years.

**Table 5:** Data analysis of QA results for the selected SLR documents by publication date

| Year | Number of studies | Average quality score | Standard deviation | Standard error |
|---|---|---|---|---|
| 2017 | 6 | 10.16 | 1.60 | 0.65 |
| 2018 | 14 | 9.64 | 1.05 | 0.28 |
| 2019 | 9 | 11.22 | 2.05 | 0.68 |
| 2020 | 17 | 10.29 | 1.68 | 0.41 |
| 2021 | 12 | 10.42 | 1.44 | 0.42 |
| 2022 | 14 | 12.28 | 2.01 | 0.54 |
| 2023 | 12 | 11.92 | 1.44 | 0.42 |
| 2024 | 4 | 11.25 | 0.96 | 0.48 |

### 3.6 Results of the SLR Selection Process

The search query stated above was utilized across specific databases, digital libraries, and search engines (i.e., Scopus, Web of Science, ACM, IEEE Xplore, and Science Direct). The initial search yielded a total of 7194 publications. Considering the substantial number of publications, we applied filters based on publication time and type. Specifically, we included publications between 2017 and 2024. It is important to note that the literature search was performed on 28 December, 2024. Therefore, only publications up to that date were considered for 2024. Applying the time and document type filters reduced the number of publications to 2315. Subsequently, a screening process was conducted by reviewing the titles and abstracts of the publications and applying the predefined inclusion and exclusion criteria. This screening step further reduced the total number of publications to 130. The selected publications in the inclusion and exclusion criteria step were eligible to read the full text and answer the eight quality assessment questions illustrated in Table 4. After applying the quality assessment and calculating the score of each article, 42 publications were excluded for not including enough information to answer the research questions of this SLR. Ultimately, 88 publications were included in this study, with 51 published as conference or proceeding papers and 37 as journal articles. Table 6 illustrates the number of publications selected for each database, search engine, and digital library during the various stages followed in this SLR. Furthermore, we integrated the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology to effectively ensure transparency

and minimise bias. Fig. 3 presents an overview of the various phases in the selection process using the PRISMA method.

**Table 6:** The number of selected papers in each step of our SLR methodology

| Databases | Initial search | Time and document type filter | Inclusion and exclusion criteria | Quality assessment |
|---|---|---|---|---|
| ACM digital library | 1664 | 583 | 25 | 18 |
| Web of science | 882 | 359 | 31 | 22 |
| Science direct | 2641 | 665 | 14 | 10 |
| IEEE xplore | 562 | 187 | 25 | 15 |
| Scopus | 1445 | 521 | 35 | 23 |
| **Total** | **7194** | **2315** | **130** | **88** |

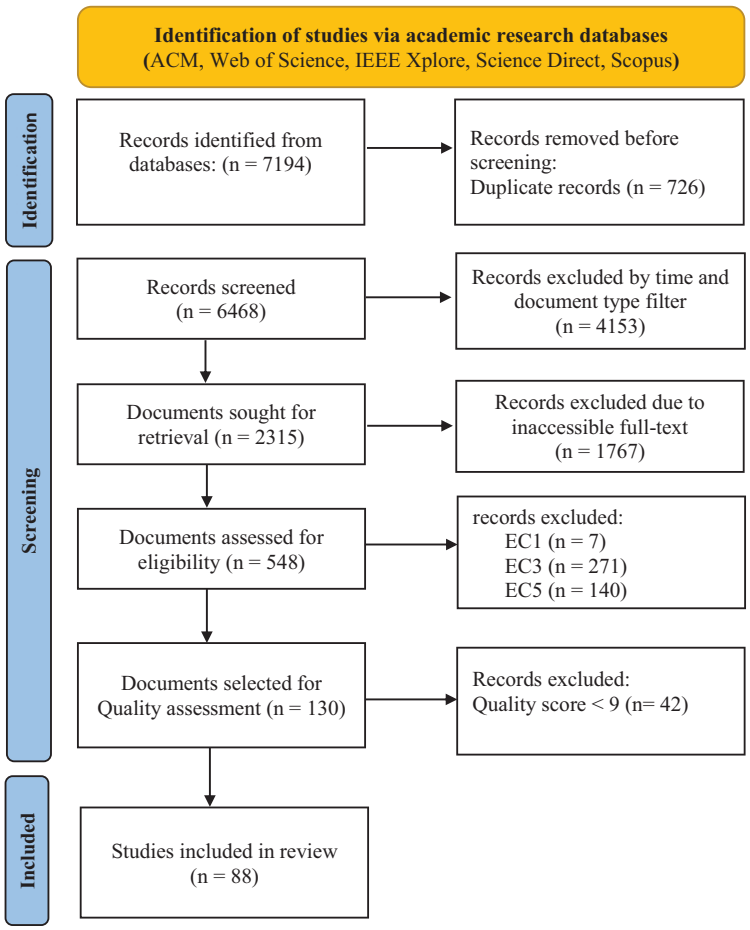


**Figure 3:** PRISMA diagram summarizing the number of documents selected in each SLR methodology phase

## 4 Results and Discussion

In this section, the results and discussion of the SLR are organized and presented in alignment with each of the previously defined research questions.

### *4.1 SSO Security Threats (RQ1)*

This section presents our analysis of the security threats on the SSO ecosystem and its standard protocols, OAuth and OpenID Connect, as highlighted in RQ1. We provide an in-depth understanding of the identified threats in the selected literature, including their potential implications and the various attacks that can exploit each one of these threats. Table 7 summarizes the findings of our analysis.

**Table 7:** SSO security threats, attacks and implications

| $ST_id$ | Security threat | Implications | Attacks |
|---|---|---|---|
| **ST1** | Online accounts abuse | Accounts takeover; PII disclosure; User impersonation; Data breach; Reputation damage. | Account Recycling attacks; Account Compromise attacks; Email Address Collision attacks; Phishing attacks; Social engineering attacks; Account enumeration attacks. |
| **ST2** | Tokens abuse and leakage | PII disclosure; Conduct malicious activities (spread malware\spam); Reputation manipulation; Tokens disclosure; User impersonation; Accounts takeover. | Eavesdropping attack; XSS attacks; Social engineering attacks; Man-in-the-middle attack; SQL injection attack; Danger-Neighbor attack; OAuth 2.0 parameter pollution attack; Path confusion attack. |
| **ST3** | OAuth2.0 and OpenID Connect incorrect implementation | Spoofing identity; PII disclosure; Tokens disclosure; Accounts takeover; Unauthorized access. | CSRF attacks; IdP Mix-up attacks; 307 redirect attack; Partial redirection URI manipulation attacks; XSS attacks; Impersonation attacks; Phishing attacks; RP account hijacking attacks; Replay attacks; Signature Bypass attacks. |
| **ST4** | Vulnerable or misconfigured SSO SDKs | Spoofing identity; Tracking user activities; Accounts takeover. | Injection attacks; CSRF attacks; Eavesdropping attack; MITM attack; Phishing attacks. |

(Continued)

**Table 7 (continued)**

| $ST_id$ | Security threat | Implications | Attacks |
|---|---|---|---|
| **ST5** | Single point of compromise | Spoofing identity; PII disclosure; Cascading account compromise; User impersonation; Data breach; Accounts takeover. | Cookie hijacking Attacks; Eavesdropping attacks; Phishing attacks; XSS attacks; Password guessing attacks; Offline dictionary attacks; Brute Force attacks; DoS and DDoS attacks; Physical access attacks; SQL injection attacks; preemptive account hijacking attack; Accounts Pre-hijacked attacks. |
| **ST6** | Fraudulent SSO authentication tickets | Spoofing identity; Accounts takeover; User impersonation. | MITM attack; Golden SAML attack; Phishing attack; Password guessing attack. |
| **ST7** | Insecure integration of dual-windows SSO | Identity theft; Account takeovers; Unauthorized access; PII disclosure; Tracking the user's activities. | Wildcard Receiver Attack; Malicious Receiver Attack; Malicious Initiator Attack; Clickjacking attacks. |
| **ST8** | Hidden active sessions | User impersonation; Accounts takeover; PII disclosure. | Physical access attacks; Session hijacking attacks; XSS attacks; CSRF attacks. |

**Online accounts abuse (ST1):** Current SSO systems rely on email addresses to link accounts with the users' real identities. The IdP is responsible for ensuring safe SSO authentication services to users by granting them access and assigning a unique UserID and other attributes to RPs. On the other hand, RPs identify the user's account associated with the provided identity by checking the UserID first, then the user's email address. Most RPs agree that a matched email address is enough to identify the user and successfully grant him access, even if the UserID is mismatched. However, Lui et al. [5] introduced, in their study, the identity-account inconsistency vulnerability, which involves a malicious action that can allow an adversary to reuse an email address in order to compromise the victim's online accounts. In such a case, the adversary impersonates the victim's identity and is considered the sole authorized user and proprietor of the email account. A systematic analysis of 100 popular RPs from Alexa's top 1000 websites reveals that 79% of RPs permit any user with the same email address to perform an SSO login to an account with an empty UserID. Additionally, 52% of the RPs allow any user with the same email address to perform an SSO login to their account, even if the UserID is different.

An attacker can abuse users' online accounts primarily through account enumeration attacks. These involve using various techniques to determine whether a specific email address is associated with an online service. By analyzing the responses of web applications to login attempts, password resets, or account creation requests, an adversary can infer the existence of an account. This can be leveraged for targeted phishing campaigns, where the attacker impersonates a service the user has an account with, increasing the likelihood that the user will fall for the attack [31]. Additionally, such information can be used by malicious entities for profiling, surveillance, or even corporate espionage. For instance, authoritarian governments may target users based on their online activities or political views, and companies may exploit this data

to identify or approach users of competing services. In practice, attackers may automate these attacks, scaling them across numerous services and email addresses. Automation allows the attacker to systematically enumerate accounts without raising suspicion, particularly when CAPTCHAs or other security mechanisms are absent [32].

**Tokens abuse and leakage (ST2):** Major websites have adopted the OAuth2.0 protocol to provide SSO authentication services to users, such as Facebook, Google, Instagram, etc. [9]. The reputation of online social networks such as Facebook is critical. People assume that information shared on these social networks is trustworthy since it has been approved and liked by a significant number of accounts. However, members of a collusion network can like other members' posts and receive likes in return in order to raise their reputation and give a fake image that they are popular. Collusion network websites require users to send OAuth access tokens to have the ability to request likes and comments. A leaked access token can begin other major security and privacy attacks in addition to reputation manipulation. Adversaries can use stolen access tokens to access users' personal information and perform malicious activities such as spreading malware [33]. Moreover, an attacker may attempt to compromise the client credentials (i.e., client_secret and client_id) by eavesdropping on the transmission during the RP authentication on the IdP process or tokens requests, by launching a SQL injection attack on the IdP database or by online guessing [34]. Then, he could masquerade as a legitimate RP and exploit the "client_id" associated privileges to gain unauthorized access to sensitive resources and obtain tokens on behalf of the RP. The attacker might employ screen-scraping techniques to mimic a user's consent and manipulate the redirect_uri to pass IdP validation checks while redirecting the authorization code (or access token in case of the implicit flow) to the malicious RP under their influence. The attacker could misuse these tokens for unauthorized access and malicious activities [13,35]. Furthermore, attackers can obtain tokens from a user's browsing history or through log files and HTTP referrers. If an attacker gains access to the user's browsing history, they can view and extract all visited URLs, including those containing tokens, thereby obtaining legitimate tokens. In addition, tokens sent via URI query parameters may be accidentally captured and stored in system log files. Similarly, if the HTTP "referer" header is used to pass information between web pages, there's a possibility that access tokens could be leaked through this mechanism [34]. Another scenario that can lead to the stealing and leakage of access tokens is exploiting the DangerNeighbor attack. This attack involves the unauthorized interception and theft of access tokens sent through postMessage by placing a malicious receiver function into the hosting page. The postMessage is a commonly used mechanism in OAuth-based SSO systems to transmit access tokens among the IdP and RP. This method is frequently employed in JavaScript SDK-based services provided by global identity providers including Facebook, Google, and LinkedIn to enable the secure transfer of access tokens between different origins. The experiment statistics, conducted by Guan et al. [36], demonstrated that 39.61% of websites using Facebook OAuth and 23.38% of websites using Google OAuth in the top 2000 Alexa websites are susceptible to DangerNeighbor attacks, leading to the compromise and leakage of access tokens.

According to the OAuth 2.0 specification (RFC 6749), the validation of redirect URIs is often performed using simple string comparison. This means that if the redirect URI provided in the authorization request matches the one registered during the client setup, it is considered valid. However, this approach can be insufficient, as it may not account for variations in the URI that could lead to vulnerabilities, such as path confusion or parameter pollution. A path confusion attack is a technique where an attacker appends specially crafted path components to a URL to exploit vulnerabilities or inconsistencies in URL parsers. This attack can target discrepancies in how various components of a web application interpret and handle URLs. By manipulating the path components, an attacker can confuse the application's routing logic, potentially resulting in unauthorized access or data exposure. In the context of OAuth 2.0 flows, attackers may exploit path confusion by registering a redirect URI that closely resembles a legitimate tenant's URI. During the

authentication process, the authorization code intended for the legitimate tenant could be mistakenly sent to the attacker's URI. The attacker can then intercept the code and exchange it for an access token, gaining unauthorized access to sensitive resources. Innocenti et al. [37] experiment reveals that, despite following the RFC-prescribed redirect URI validation strategy, 6 out of 16 identity providers (IdPs) failed to properly validate the redirect URI, leaving them vulnerable to path confusion attacks. An OAuth 2.0 parameter pollution attack (OPP) is a specific application of the broader HTTP parameter pollution (HPP) technique, targeting OAuth 2.0 flows. In an HPP attack, an attacker sends a request containing multiple parameters with identical names but different values. Since different components in a web application may handle such parameters inconsistently, this can lead to unexpected or insecure behavior. In the context of OAuth 2.0, OPP occurs when an attacker injects multiple "code" parameters into the OAuth authorization flow. Specifically, after the user authenticates, the attacker attempts to influence the process so that the RP receives two "code" parameters: one legitimate value and another malicious one injected by the attacker. Due to inconsistencies in how IdPs process query strings, the system might accept both parameters, which can result in the attacker bypassing security checks and gaining a valid access token. Similarly, the results of the experiment conducted by Innocenti et al. [37] demonstrate that 10 out of 16 IdPs were affected by OPP attacks.

**OAuth2.0 and OpenID Connect incorrect implementation (ST3):** SSO services are mostly based on OAuth 2.0 and OIDC protocols. Previous studies reveal that the incorrect implementation of SSO protocols may result in significant security vulnerabilities due to RP developers' mistakes and complex IdP implementation guidances [36,38–40]. One of the most common vulnerabilities found in many actual implementations of OAuth 2.0 and OpenID Connect is Cross-site request forgery (CSRF). A study conducted in 2016 analysed the top 500 websites in the US and Chinese and found that 61.23% of the 405 websites that support OAuth2.0 did not use the parameter state, which is considered the proposed standard's countermeasure to prevent CSRF attacks in OAuth [41]. Even worse, 55.31% of websites that employ the state parameter remain vulnerable to CSRF attacks due to this parameter being misused or improperly handled [42]. A CSRF attacker can gain control of a victim user's account without knowing their username and password, instead by making the victim access RP resources on their behalf. In 2020, another experiment was carried out to assess the prevalence of CSRF vulnerabilities in OAuth implementations on the 395 Alexa high-ranked sites [43]. The study revealed that approximately 36% of the targeted sites are still susceptible to CSRF attacks. Additionally, 20% of the less easily detected vulnerable sites utilize novel attack strategies, indicating a significant prevalence of this security threat in OAuth implementations. These strategies include manipulating the value of the state parameter contained within the authorization URL by replacing it with an empty string, replacing the state parameter value with a randomly generated string, completely removing the state parameter from the authorization URL, or sending the URL unchanged to the victim if it didn't originally have a state parameter. The study conducted by Sumongkayothin et al. [44] analyzed more than 45 websites that utilize OAuth and revealed that a notable 84.4% of these websites lack essential parameters, leading to potential security vulnerabilities. In 2023, Westers et al. [45] have analyzed 3020 SSO logins using their developed tool, SSO-MONITOR, to detect the security and privacy flows of current SSO implementations. The analysis results reveal that 447 logins had entirely no protection against CSRF attacks, and 337 cases were found to have weak CSRF protection.

Furthermore, the inadequate validation of tokens and the authenticity of their senders pose a substantial security risk to the OAuth2.0 and OIDC protocols. An attacker can impersonate the victim user and gain complete access to his RP account by using a malicious RP to get an access token from the IdP [46]. Since the access token is a bearer token that can be utilized by any RP that owns it, and because of the improper validation of the RP identity and confusion between authorization and authentication, the attacker can gain access to the user's PII in the IdP by only providing the malicious RP access token [15]. If an IdP

uses the 307 HTTP status code to redirect the user's browser back to the RP, an attacker can get the user credentials in the IdP [39]. Moreover, it is crucial to highlight that in the core OAuth 2.0 and OpenID Connect specifications, the authorization code, access token, ID token, and refresh token must be sent over a secure channel using TLS. Otherwise, an attacker may attempt to eavesdrop, intercept and manipulate the tokens during transmission [47]. By obtaining valid tokens, the attacker may attempt to manufacture a forged token, make changes to an existing valid token, or even replay a valid token to impersonate the victim and gain unauthorized access to protected resources [13,35]. In addition to the insecure transmission channel, improper validation of these tokens and their sender authenticity can exacerbate the security weakness of these protocols [10]. Another class of attacks works against RPs supporting multiple IdPs, called Partial Redirection URI Manipulation PRURIM attacks. An attacker of this class can obtain the code without the victim user's awareness, and use it to get full access to his RP account because of the improper validation of the redirect_uri. If the IdPs solely validate the origin part of the redirect_uri (i.e., that specifies the RP) when receiving an authorization request, rather than the complete redirect_uri, an attacker can manipulate a portion of the redirect_uri and embed an unauthorized request in an iframe or image on the target RP, or exploit XSS vulnerabilities, to get an authorization code from the legitimate IdP [48]. An attacker with a malicious IdP can then leverage the stolen code for a variety of malicious actions, including defrauding users and spoofing their identities [49]. Qiu et al. [50] have conducted a study on 500 Chinese websites supporting SSO and found that 59.66% of the OAuth-based SSO websites are vulnerable to RP account hijacking attacks exploiting the vulnerability arising from the modification of the redirect_uri. This vulnerability allows unauthorized parties to modify the redirect_uri during the authorization process to get a valid authorization code. The attacker can utilize the stolen code to obtain an access token and retrieve the victim user's ID from a target IdP by initiating a new login procedure at an RP, intercepting the authorization response redirection and replacing the code with the stolen one to retrieve the access token.

**Vulnerable or misconfigured SSO's SDKs (ST4):** Popular IdPs such as Google and Facebook provide various Software Development Kits (SDKs) to simplify the integration of SSO services in RPs. To increase flexibility, RP developers can integrate different SDK modules to support multiple IdPs at the same time [49]. SSO SDK implementation can lead to a significant security and privacy risk if the source code of these SDKs is vulnerable or if developers do not follow the documentation correctly [51]. If an attacker manages to set up a malicious RP and deceive a user into logging into it, they can exploit vulnerabilities in the SSO SDK to remotely manipulate the user's accounts and monitor their activities across any RP. In some cases, this can even result in a complete takeover of the user's legitimate RP accounts. Developers must ensure the security and integrity of the SDKs they integrate and follow the recommended guidelines to mitigate these risks effectively.

**Single point of compromise (ST5):** In SSO systems, the IdP is considered a single point of failure and an obvious compromise target. If an attacker gets to compromise an IdP account, he will be able to completely access and take control of all the RP accounts linked with that IdP, discover their master secret passwords, and retrieve saved hashed passwords for use in offline dictionary attacks [7,40,52–54]. Furthermore, due to the seamless integration benefits of SSO, some services can behave as identity providers enabling account and identity management and at the same time as relying parties that allow users to log in with other identity providers. Ghasmisharif et al. [16] found that 52% of Wikipedia's top 65 IdPs supporting OAuth and/or OpenID Connect protocols behave both as IdPs and RPs. However, this feature increases SSO security risk since it augments the attack surface. For example, a hijacked Facebook account can lead to the compromise of 226 additional RP accounts in the Alexa top 100K if the attacker focuses first on compromising RPs that are IdPs for other RPs. Generally, an attacker can compromise an IdP account using two different scenarios: compromised IdP passwords or hijacked session cookies [16,49,55]. A web attacker can obtain

an IdP password using phishing techniques, which is considered the most common reason for compromise even in major IdPs. Once the attacker gains the user's IdP password, he completely takes over his IdP account and all the RP accounts supporting this IdP. Moreover, an attacker who has access to a victim's IdP account might utilize SSO to proactively establish an RP account that the victim user does not already have and then patiently await for him to join the service. After the creation manual of this attack on 95 RPs, Ghasmisharif et al. [16] found that users will not be suspicious of this attack since the "login" and "register" options redirect users to the same point and the attacker will have the same level of access as an RP account-hijacking attacker. The problem here is the IdP legitimate user will get email alerts for each RP account that is created. To deal with this, the attacker can use the hijacked IdP password to add his own email and set it as the IdP account principal email, create the preemptive RP accounts, and delete his email from the user's IdP account to not leave any trace. Along with the preemptive account hijacking attack, Sudhodanan and Paverd [56] demonstrated that there is a whole class of similar attacks, called accounts Pre-hijacked attacks, that exploit vulnerabilities in the account creation process to gain unauthorized access to user accounts. These attacks are; unexpired email change attack, unverified email change attack, unverified phone number change attack, unverified email addition attack, and unverified phone number addition attack.

Furthermore, an attacker can completely take over IdP accounts and obtain the same level of access and control as a legitimate authenticated user by using hijacked session cookies [57]. A passive eavesdropping attacker can intercept and collect session cookies exposed over unencrypted HTTP connections. By default, even if a domain supports HTTPS, browsers attempt to send requests to the domain over HTTP before the web server redirects the requests to HTTPS [58]. If the domain does not support the HTTP Strict Transport Security (HSTS) mechanism and the cookies are not protected with a secure flag, the attacker can retrieve session cookies in clear text in the initial browser HTTP request [16,59]. The attacker can also access session cookies by executing malicious Javascript code within the domain's origin via XSS attacks or by running malicious third-party scripts within the web app's origin if it allows including scripts without being isolated in an iframe [59]. To examine the feasibility of hijacked RP accounts and the level of access a hijacked cookie attacker can have, Ghasemisharif et al. [16] conducted an experiment to evaluate the attacker's access level using hijacked IdP cookies on 29 websites of the Alexa top 500 supporting Facebook SSO, including reading messages, sending new messages, ordering items, adding new posts, etc. As a result of this experiment, they found that the attacker has an identical access level as a legitimate authenticated user. Of the 29 websites only one (i.e., the Guardian) requires the attacker to re-authenticate over SSO by entering the Facebook password to access the settings option. To make matters worse, the victims did not receive any warning during this experiment, and even if he tried to check the list of recent sessions, the attacker's sessions would not appear in the list unless they lasted more than an hour. Furthermore, even after losing access to the victim's IdP account, an attacker can still access RP accounts in the long term by replacing the victim's email address at the RP with his own email and then setting or resetting the password associated with the RP [60]. The test results of how the 29 web RPs react in this scenario show that the attacker can sustain long-term access in 22 RPs out of the 29 without knowing the user password. Furthermore, If the IdP goes down or becomes unavailable, it disrupts the authentication process for all the underlying users, impacting their ability to access any services that require authentication. This dependency on the IdP's availability creates a single point of failure, as the entire SSO system's functionality is contingent on the uninterrupted operation of the IdP [61].

**Fraudulent SSO authentication tickets (ST6):** SSO allows users to authenticate themselves once and have access to numerous apps or services without the need to re-enter their credentials. Within this context, the IdP generates SSO tickets, such as ID tokens in OpenID Connect, to authenticate users to an RP. These SSO tickets are digitally signed by the IdP to ensure their legitimacy. The RP must verify and validate these tickets to authenticate the ticket's bearer and grant him access to the RP's resources [62]. However, recent

security incidents have highlighted that even well-protected signing systems, such as certification authorities (CAs), can be compromised to sign fraudulent messages. The IdP, being a single point of compromise in the SSO system, is particularly even more exposed to vulnerabilities and attacks compared to CAs. A malicious or compromised IdP can generate fraudulent or fake SSO tickets for a victim user account, enabling adversaries to seamlessly access all associated applications without triggering any error notifications or alarms [63]. Another contributing factor to the success of this threat is the improper or incomplete validation of SSO tickets at the target RP [49]. Upon gaining legitimate SSO tickets upon a compromised IdP or using attacks like man-in-the-middle, an adversary can trick an RP into successfully validating a fraudulent SSO ticket by modifying certain parameters in the ticket. For example, in an ID token, the attacker may modify parameters such as the "aud" parameter to set the right token recipient or the "exp" parameter to prevent rejection caused by token expiration. However, these changes may trigger the RP's signature verification process, which could alert the RP about problems with the ticket's integrity and lead to ticket rejection. To overcome this, the adversary can use Bypassing signature techniques, manipulate the key used for signature verification, and in some cases where RPs have a policy in place to avoid denial-of-service to end-users, remove the signature and make the RP accept fake tickets without signature [49]. Moreover, an attacker can use a temporary SSO ticket that was generated for the hacked IdP account to create a long-term token. For example, the attacker may set a persistent cookie at the RP, allowing ongoing access to the victim's account [63]. The improper validation of SSO tickets and their signature may enable an adversary to achieve his goals using a malicious IdP to issue a fresh SSO ticket created from scratch containing legitimate values of the legitimate entities (i.e., the IdP, end-user, and RP).

**Insecure integration of dual-windows SSO (ST7):** In addition to standard protocols like OAuth 2.0 and OpenID Connect, SSO can use another approach, called dual-window SSO to authenticate a user on an RP using his identity on an IdP. Dual-window SSO uses iframe and popup flows, which rely on a technique called In-Browser Communication (InBC). The InBC uses the postMessage technique to exchange tokens securely between various windows. Instead of sending the token through the URL, it is passed between windows using a messaging mechanism provided by the browser. Jannett et al. [64] investigation reveals that 273 websites (27%) of the Tranco top 1k list provide SSO login, on which 153 websites (56%) support dual-window SSO. However, dual-window SSO is vulnerable to three major security attacks; Wildcard Receiver Attack (WRA), Malicious Receiver Attack (MRA), and Malicious Initiator Attack (MIA), which impact the confidentiality and authenticity of SSO exchanged messages. We can trace back all attacks to the strong reliance on manual integrations rather than SDKs for implementing InBC techniques. The security analysis of the Apple, Facebook and Google SDKs' InBCs demonstrates their security and efficiency. The evaluation demonstrates that 77 (50%) out of 153 dual-window SSO-based websites use manual integration. The results of the security analysis of these 77 websites using the DISTINCT tool reveal that 24 (31%) of the websites, which have manually integrated dual-window SSO, do not properly implement InBCs [64].

**Hidden active session (ST8):** This threat refers to the situation where users are unaware of the existence of active IdP sessions along with RP sessions while using SSO services. When users sign out of an RP service, they assume that all active sessions, including the IdP session, are terminated. However, in reality, the active IdP session remains in the user's browser, potentially leading to unintended and unauthorized access to the user's resources and sensitive information [65,66]. For example, if a user logs out of an application in a public library but leaves the active IdP session open, a malicious actor could exploit this opportunity to access the user's account and data. Additionally, hidden active sessions can lead to confusion and frustration for users who may assume that signing out of one service terminates all active sessions. However, in the case of actual SSO implementations, the service remains accessible without the need for user credentials and re-signing in with the IdP as long as the IdP session is still active [66].

### *4.2 SSO Privacy Threats (RQ2)*

Single Sign-On (SSO) provides convenience and enhances user experiences by enabling individuals to access multiple applications with a single set of credentials. However, this convenience often comes at the expense of privacy. This section delves into the privacy concerns associated with SSO discussed in the literature, as highlighted in RQ2. Table 8 outlines the primary SSO privacy threats, their potential impacts and the attacks capable of exploiting each of these threats.

**Table 8:** SSO privacy threats, attacks and implications

| $PT_id$ | Privacy threat | Implications | Attacks |
|---|---|---|---|
| **PT1** | Unauthorized data collection and sharing | Violations of privacy; Unauthorized data sharing; Unauthorized disclosure of sensitive data; Misuse of user data. | Phishing attack; Session Hijacking attack; CSRF attacks. |
| **PT2** | Tracking and profiling users | Unauthorized access; Privacy intrusions; Loss of control over users' PII. | Phishing attack; Social engineering attacks; Replay attack; Cookie-hijacking attack. |
| **PT3** | Consent manipulation | Unwittingly granting excessive permissions; Violations of privacy; Unauthorized data sharing; Misuse of user data. | Social engineering attacks. |
| **PT4** | Account linking | Loss of user anonymity; Enabling profiling. Violations of privacy; Enabling tracking. | CSRF attacks; MITM attack; Phishing attack. |
| **PT5** | User PII leakage | Unauthorized disclosure of user's sensitive data; Violations of privacy; Potential identity theft. | MITM attack; Session Hijacking attack; XSS attacks; Spoofing attacks. |

**Unauthorized data collection and sharing (PT1):** This threat revolves around the broad access that users may inadvertently provide to a centralized identity provider or authentication service. When users connect multiple services to a single SSO provider, it consolidates user data in one location [67]. While this consolidation can enhance user convenience, it also raises concerns about potential data sharing and access to sensitive information. If not carefully managed, data sharing can lead to privacy breaches, unauthorized data access, and the exposure of sensitive user information to third-party applications [68,69]. In a typical SSO ecosystem, IdPs are expected to provide minimal information about the user's identity to the RPs, usually in the form of security tokens (e.g., Assertion or JWT). However, major IdPs like Google and Facebook allow RPs to access account data and provision long-term access tokens, even when the user is signed off [70,71]. This could result in unnecessary exposure of sensitive user information to RPs that do not require access to such data. Moreover, curious IdPs might be tempted to store more personal attributes than necessary for various purposes, such as tracking user behaviour, profiling users, or selling data to third parties [72]. This could raise privacy concerns as users may not be fully aware of the extent of data collection and sharing. For instance, the OAuth 2.0 and OpenID Connect SSO systems often collect and store user data, including login credentials, browsing history, purchase history, and other behavioural information without user consent, leading to privacy violations and breaches of trust [73]. As a result, user consent and control over data become essential aspects to address in such cases [6,49,74–77].

**Tracking and profiling users (PT2):** Tracking and profiling users in the context of SSO refers to the practice of gathering user data and tracking their activities to create a detailed profile of their habits, behaviour, and preferences [59,70,78]. When users connect to different services using SSO, the IdP can potentially track their activities across these services, creating a more comprehensive profile of user behaviour and preferences [79–81]. This tracking can be facilitated by profiling cookies used for targeted advertising or other purposes [49,68,69]. As a result, users may be subjected to targeted advertising, profiling, and tracking without their consent, resulting in privacy intrusions and a loss of control over their PII. Furthermore, users' lack of awareness regarding privacy settings, access rights, and sharing scope when providing consent for the information shared with RPs during SSO authentication exacerbates these risks and further compromises their privacy [75,76]. Therefore, it is crucial to educate users about privacy settings, encourage them to review and adjust their preferences, provide opt-out options for personalized advertising, and implement transparency measures to inform users about how the collection, use, and sharing of their data occurred [71,82].

**Consent Manipulation (PT3):** emerges as a notable threat within SSO systems, introducing privacy challenges in the management of user consent. Users must possess clear and granular control over the data they share and with whom. However, certain SSO implementations may fall short of providing users with sufficient visibility or control over data-sharing permissions [63,67,74,83]. Exploiting this vulnerability, attackers may engage in manipulative practices, attempting to deceive users into granting excessive permissions to applications through fraudulent user interfaces. The impact of Consent Manipulation is significant, as users may unwittingly authorize applications to access more data than intended, leading to unintended data sharing and the potential misuse of personal information [49,75]. To counteract this threat, robust defence solutions are essential. This includes designing user interfaces that transparently present data access requests and permissions, implementing user-friendly consent management mechanisms, and fostering user awareness regarding the risks associated with SSO [78,84]. In a 2023 experiment, Westers et al. [45] identified 200 instances of compromised user privacy among the 3020 analyzed SSO logins. In these cases, user information was shared between clients and identity providers without the user's consent, enabling user tracking without their awareness.

**Account Linking (PT4):** presents a significant threat in the context of SSO, where RPs may use mechanisms like browser fingerprinting or collecting Personally Identifiable Information (PII) such as email addresses to establish links of user identity across multiple RP accounts [70]. This linkage can result in a loss of user anonymity and privacy, as the user's identity and activities on one platform become intertwined with those on others. Exploiting this interconnection, malicious actors may carry out CSRF attacks, deceiving users into unintentionally performing actions on other platforms [67,68,77]. The impact of the account linking threat is significant, as it compromises user anonymity and enables undesired connections between user accounts. This poses privacy concerns and creates opportunities for profiling and tracking [79–81]. Therefore, it is crucial to empower users with transparent and precise control over which accounts link to their accounts, enabling them to make informed decisions about their interconnected identities.

**User PII Leakage (PT5):** Misconfigured or incorrectly implemented SSO systems can potentially expose user information to third-party apps or services without obtaining the user's consent [46,70,81]. This threat can happen through both privileged and non-privileged access [6,76]. While IdPs and RPs are generally considered trustworthy agents, attackers may possess different levels of access to their infrastructures [36,49]. This is because not all employees or stakeholders within an organization can be fully trusted, despite the overall trustworthiness of the organization itself [49]. Moreover, if confidential data is communicated without adequate encryption across insecure communication channels, malicious individuals can intercept

the conversation between the user and the SSO provider to access sensitive user data [69]. This can lead to several negative impacts, such as unauthorized disclosure of sensitive user data, privacy violations, and identity theft.

### *4.3 Threat Mitigation Measures (RQ3)*

In this section, we will address the third question (RQ3) of our literature review by identifying countermeasures that can effectively mitigate the SSO security and privacy threats identified in the Sections 4.1 and 4.2. Fig. 4 presents a summary of the different defence measures that can be implemented for each of the SSO security and privacy threats.

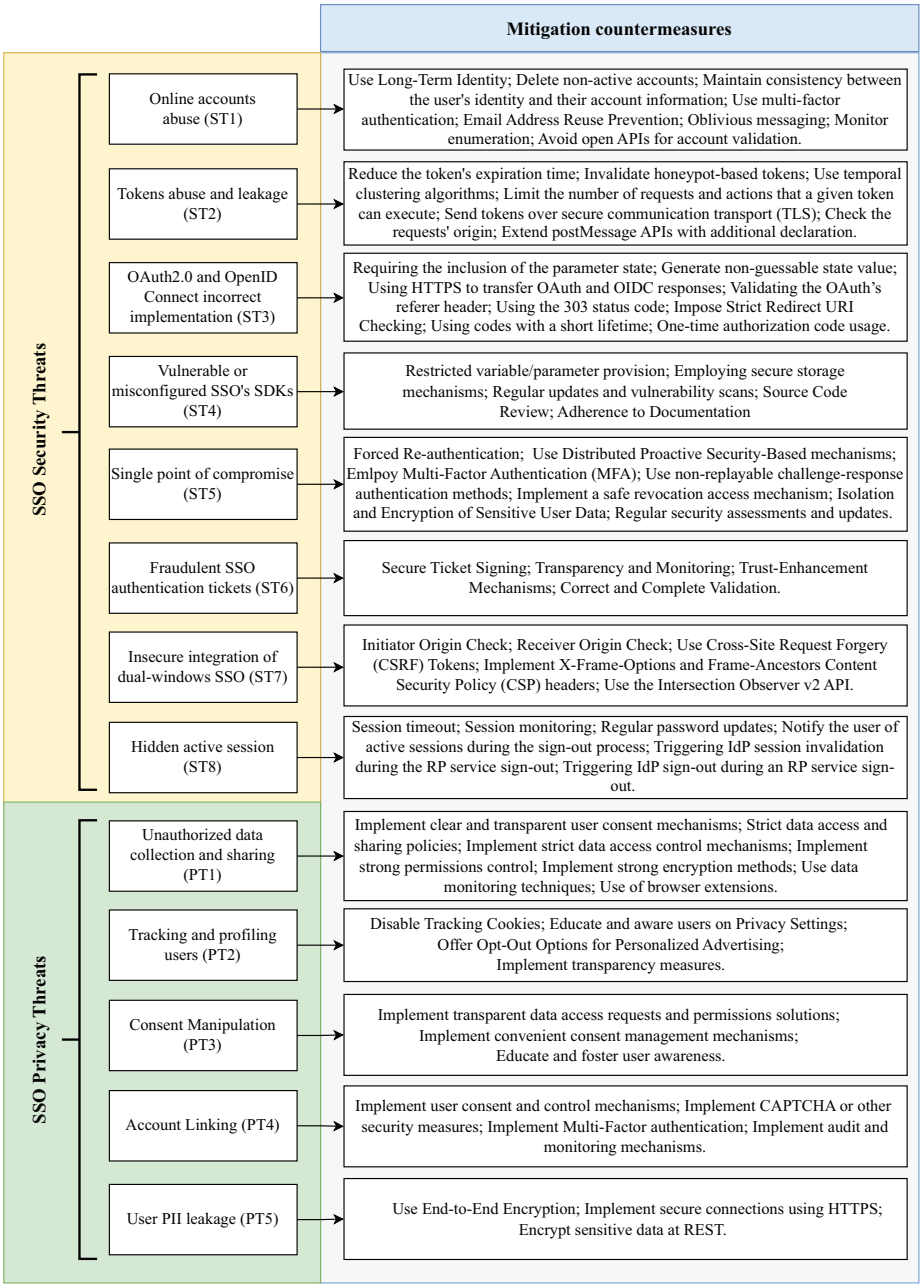


**Figure 4:** Mitigation countermeasures to SSO security and privacy threats

*4.3.1 Mitigating Security Threats*

**Online accounts abuse (ST1):** To mitigate this threat, relying on using only long-term identity to establish online accounts can help to reduce the risk of inconsistency between identity and accounts. This means that the identity used to create an account should be stable and not subject to frequent changes. Additionally, removing non-active accounts ensures that unnecessary personal data is not retained in the system, reduces security risks, and ensures that the accounts within the system accurately reflect the current status of users' identities [5]. RPs should prompt users to update their attributes when they gain access to an account with a matching userID. This helps maintain consistency between the user's identity and their account information. Using multi-factor authentication provides an additional layer of security that significantly reduces the risk of identity-account inconsistency by making it more challenging for unauthorized individuals to access user accounts. Additionally, IdPs should not allow the reuse of email addresses to prevent the reassignment of an email address. Furthermore, RPs should display generic messages for operations like password reset, login, or account creation. For example, using messages like "If an account exists, a password-reset link has been sent" prevents attackers from determining whether an account exists based on system responses. Additionally, detecting multiple requests originating from the same IP address or linked to different email addresses can alert RPs to account enumeration attacks. Subsequently, RPs should avoid offering APIs that allow attackers to verify whether an account exists based on input parameters [32].

**Tokens abuse and leakage (ST2):** To mitigate this threat, shorting tokens' time expiration can minimize the window of opportunities for potential abuse. Additionally, using Honeypot-based tokens can help to detect and track unauthorized access. When an unauthorized user attempts to use a honeypot-based token, it triggers an alert, allowing the system to identify and respond by revoking access to these tokens when they are used in unauthorized or suspicious ways, thereby preventing potential misuse and protecting the system from security breaches [9,33]. Moreover, using temporal clustering algorithms can detect abnormal patterns of token usage over time, allowing the identification and mitigation of potential threats. Also, utilizing Transport Layer Security (TLS) for token transmission ensures that the data is encrypted and protected from interception or tampering. Limiting the number of requests and actions that a given token can execute can minimize the risk of abuse. Furthermore, checking the origin of requests via a JavaScript function wrapper helps validate the authenticity of token-based requests, reducing the risk of unauthorized usage. Additionally, the extension of postMessage APIs [85] with an additional declaration on the designated receiver function enhances the security of token-based communication and reduces the likelihood of leakage or abuse [13,36].

**OAuth2.0 and OpenID Connect incorrect implementation (ST3):** mitigating this threat requires including the state parameter in the authorization request and response and using it correctly. Additionally, RP should generate an unpredictable and unique state value to ensure security. Using HTTPS to transfer OAuth and OIDC responses reduces the risk of interception and tampering. The OAuth's referer header should point to either the RP or IdP domains to validate the request's origin and prevent unauthorized redirection [36,38,39]. Furthermore, employing the 303 status code to redirect the user's browser to the RP and imposing strict Redirect URI Checking, also reduce the risk of unauthorized redirects. Using codes with a short lifetime helps limit their potential exposure and reduces the window of opportunity for malicious actors to misuse them. Additionally, Authorization codes should be designed for one-time usage, preventing replay attacks and unauthorized reuse of the codes [17,48,86].

**Vulnerable or misconfigured SSO's SDKs (ST4):** To mitigate this threat, SDK developers should carefully filter which variables/parameters can be provided to users. Restricting access to certain elements can minimize the risk of unauthorized manipulation or exploitation of the SDK. Securing storage using hashes or

other techniques can help protect sensitive data within the SSO SDKs. Furthermore, it is essential to regularly update SSO SDKs and conduct thorough scans for possible vulnerabilities. This enables organizations to identify and address potential security flaws before they can be exploited. Additionally, it is crucial to thoroughly review the source code of the SDK before integration in order to identify and address any potential vulnerabilities or misconfiguration. Also, carefully following the documentation provided by the SDK provider ensures that the integration and usage of the SDK align with best practices and security recommendations, reducing the likelihood of misconfiguration or vulnerability [9,48,49].

**Single point of compromise (ST5):** Mitigating this threat involves implementing a policy to force the user to re-authenticate at the beginning of each session, thereby preventing unauthorized access if the current session is compromised. Additionally, using distributed proactive security-based solutions, such as distributed or threshold-based authentication mechanisms, can mitigate the risks associated with this threat. Furthermore, the implementation of multi-factor authentication adds an extra layer of security. Additionally, employing non-replayable challenge-response authentication methods can help to prevent replay attacks and unauthorized access attempts. A safe revocation access mechanism can promptly revoke access for compromised or unauthorized entities, limiting the impact of a potential compromise. Moreover, isolating and encrypting sensitive user data within the IdP protects it from unauthorized access in the event of a compromise. Additionally, conducting regular security assessments and updates for the IdP is essential for identifying and addressing potential vulnerabilities, thus ensuring that the system remains resilient against evolving threats [49,59–61,70].

**Fraudulent SSO authentication tickets (ST6):** Mitigating this threat requires several countermeasures. SSO tickets should be signed with a multi-key or a fragmented key to enhance their security and integrity, making it more difficult for malicious actors to forge or tamper with the tickets. Furthermore, ensure ticket transparency using public logs to monitor IdP operations. This provides visibility into the operations of IdPs, allowing for the detection of any suspicious activities or unauthorized ticket issuance. Additionally, implementing trust-enhancement mechanisms such as pinning, public logging, multi-path verification, user-controlled policies, restricted scopes of services, and multi-authority certification can bolster the overall security and trustworthiness of the SSO authentication process. Moreover, SSO tickets must be correctly and completely validated to ensure that only legitimate and authorized tickets are accepted for authentication, reducing the risk of fraudulent ticket usage [49,62,63].

**Insecure integration of dual-windows SSO (ST7):** Mitigating this threat requires implementing a thorough check of the initiator's origin to ensure that the request is originating from an authorized and legitimate source. Similarly, perform a comprehensive check of the receiver's origin to verify the legitimacy of the incoming request and prevent unauthorized access. Additionally, the use of CSRF tokens ensures that requests originate from authorized and authenticated sources. Implementing X-Frame-Options and Frame-Ancestors Content Security Policy (CSP) headers controls how the SSO integration is embedded within frames and mitigates the risk of clickjacking and other related attacks. Furthermore, utilizing the Intersection Observer v2 API [87] efficiently manages and observes intersections between elements in the dual-windows SSO integration, enhancing both its security and performance [64].

**Hidden active session (ST8):** Several effective countermeasures can be used to mitigate this threat. Session timeout mechanisms, which automatically log out users after a period of inactivity, are instrumental in mitigating the risk of hidden active sessions. Regularly monitoring active sessions and providing administrators with tools to view and manage these sessions can significantly reduce the risks associated with this threat [88]. Enforcing regular password updates can help prevent unauthorized access and reduce the likelihood of hidden active sessions. Notifying the user of active sessions during the sign-out process is crucial for making them aware of any ongoing sessions and allowing them to take appropriate action. Additionally,

triggering IdP session invalidation during the RP service sign-out ensures that all active sessions associated with the user are invalidated when they sign out from a specific service. Moreover, automatically triggering IdP sign-out during an RP service sign-out guarantees that the user is logged out from the Identity Provider when they sign out from a relying party service. These measures collectively contribute to a more robust defence against the hidden active session threat [65,66].

*4.3.2 Mitigating Privacy Threats*

**Unauthorized data collection and sharing (PT1):** To mitigate this threat, organizations should implement clear and transparent user consent mechanisms that fully inform individuals about the collection and sharing of their personal data and allow them to provide explicit consent. Systems should also allow users to customize the permissions they grant to IdPs, enabling them to opt out of sharing specific types of data while still using SSO [59,67]. Implementing advanced security measures, such as encryption and continuous monitoring, further protects sensitive data from unauthorized access and sharing. Enforcing strict data access and sharing policies within an organization ensures that data is only accessed and shared in accordance with established guidelines and regulations [36,89]. Develop and promote browser extensions that can alert users to known vulnerabilities in RP and IdP implementations, helping them make informed decisions about which login options to choose [90,91].

**Tracking and profiling users (PT2):** To mitigate this threat, it is imperative to disable tracking cookies to help prevent unauthorized tracking of users' online activities and behaviours. Additionally, educating users on privacy settings and encouraging them to review and adjust their preferences empowers them to take control of their online privacy and limit the extent to which they can be tracked and profiled [75,76]. Furthermore, providing users with opt-out options for personalized advertising. This means giving users the choice to decline or opt out of having their online activities used for targeted advertising purposes, which gives them greater control over their online privacy Additionally, implementing transparency measures provides users with clear and comprehensive information about how their data is being collected, tracked, and utilized for profiling and targeted advertising [49,59].

**Consent manipulation (PT3):** To effectively mitigate this threat, it is crucial to implement robust defence solutions. This involves designing user interfaces that transparently present data access requests and permissions, enabling users to make informed decisions about granting consent for their data to be accessed or used [92]. Additionally, implementing user-friendly consent management mechanisms allows users to easily review and adjust their consent preferences, providing them with greater control over how their data is utilized [49,75]. Furthermore, educating and fostering user awareness about data privacy and consent practices empowers individuals to understand the implications of their consent decisions and make informed choices regarding the use of their sensitive data [78,84].

**Account linking (PT4):** Mitigating this threat involves implementing effective defence solutions. Providing users with transparent and granular control over which accounts are linked is crucial, allowing them to make informed choices about their interconnected identities. Additionally, implementing security measures such as CAPTCHA helps prevent automated account linking, adding an extra layer of protection against potential threats associated with Account Linking. Furthermore, implementing Multi-factor authentication to verify the identity of users attempting to link their accounts. Maintaining audit trails and regularly monitoring account linking activities help detect any suspicious or unauthorized attempts, thereby identifying and mitigating potential threats [49,89].

**User PII leakage (PT5):** To mitigate this threat, several countermeasures can be used. Organizations can use end-to-end encryption to protect data in transit. This ensures that sensitive information remains

encrypted and secure as it travels between users and servers, reducing the risk of interception and unauthorized access [49,59]. Additionally, secure connections using HTTPS should be implemented to establish a secure and encrypted communication channel between users and web servers [47]. This helps prevent eavesdropping and tampering with data during transmission. Furthermore, encrypt sensitive data at rest to safeguard stored information. By encrypting data at REST, organizations can protect user PII from unauthorized access in the event of a data breach or unauthorized system access [44,63,93].

### *4.4 State of the Art and Research Gaps in SSO Schemes (RQ4)*

In this section, we will answer the fourth research question (RQ4) of our SLR by examining the state of the art of the existing SSO schemes proposed in the literature and what are the security and privacy threats these schemes aim to mitigate. This analysis will enable us to pinpoint the specific areas within security and privacy that require improvement or further research. Based on the security and privacy threats identified in Sections 4.1 and 4.2, Table 9 illustrates 47 SSO schemes proposed in the selected documents of our SLR. Furthermore, the table evaluates the coverage level of each identified SSO threat across these analysed SSO schemes, representing the proportion of schemes that address a given threat in the literature. This coverage level is calculated as the ratio of schemes mitigating a specific threat to the total number of schemes analyzed (i.e., 47 SSO schemes).

**Table 9:** Coverage level of SSO security and privacy threats by existing schemes in the literature

| SSO | SSO threats | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Schemes | ST1 | ST2 | ST3 | ST4 | ST5 | ST6 | ST7 | ST8 | PT1 | PT2 | PT3 | PT4 | PT5 |
| [76] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ○ |
| [36] | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| [39] | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [75] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ○ | ○ |
| [52] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [94] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [64] | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ● |
| [95] | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [96] | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [97] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ |
| [63] | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [81] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● |
| [69] | ○ | ● | ○ | ○ | ● | ● | ○ | ○ | ○ | ● | ○ | ○ | ● |
| [73] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● |
| [65] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● |
| [51] | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [16] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ |
| [17] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ● | ● |
| [98] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ○ |
| [99] | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| [100] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ● | ● |
| [101] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| [80] | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ● | ○ | ● | ● |
| [7] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| [68] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● | ○ | ● | ○ | ● |
| [102] | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ○ | ● |
| [103] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ |
| [104] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ● | ● | ○ | ○ | ● |
| [105] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● |
| [10] | ○ | ● | ○ | ○ | ● | ● | ○ | ○ | ○ | ● | ○ | ● | ● |
| [106] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ○ | ○ |
| [107] | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ● | ○ | ● |
| [108] | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [109] | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

(Continued)

**Table 9 (continued)**

| SSO Schemes | SSO threats | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ST1 | ST2 | ST3 | ST4 | ST5 | ST6 | ST7 | ST8 | PT1 | PT2 | PT3 | PT4 | PT5 |
| [110] | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [74] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ● | ● | ● | ○ | ○ |
| [111] | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| [82] | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [44] | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [90] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ |
| [79] | ○ | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| [112] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● | ● | ● | ● | ● |
| [113] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ● | ● | ● |
| [114] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● | ○ | ○ | ● | ● |
| [115] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ○ |
| [116] | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● | ● | ○ | ○ | ● |
| [117] | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ● | ● | ○ | ○ | ● |
| **Coverage level (%)** | **2.13** | **36.17** | **12.76** | **2.13** | **42.55** | **17.02** | **2.13** | **6.38** | **40.42** | **38.30** | **29.79** | **19.15** | **51.06** |

Note: The bullet (●) indicates that the identified SSO Schemes considers the specified SSO Threats in the corresponding column, while the circle (○) indicates otherwise.

Referring to the results of our analysis, we noticed that Online accounts abuse (ST1), Insecure integration of dual-windows SSO (ST7) and Vulnerable or misconfigured SSO's SDKs (ST4) threats are the less addressed in the literature with a coverage level of 2.13%. Bilal et al.'s scheme [99] stands out as the sole work that has addressed the concern of user identity and account inconsistency, and this is achieved through a multistep authentication process relying on email authenticity and a Special Secret Encrypted Alphanumeric String (SSEAS). Similarly, Jannett et al. [64] were the only study that tackled the insecure integration of dual-window SSO threats by developing the DISTINCT tool. DISTINCT performs a dynamic analysis of the JavaScript code running within the SSO flow, transforming it into a sequence diagram to illustrate all communicating entities and their exchanged messages. Finally, it highlighted insecure communication channels and assessed the severity of quantifying flows in a dual-window SSO. Likewise, vulnerable or misconfigured SDKs threat is solely addressed by Yang et al.'s work [51], and this by developing an automated testing tool called S3KVetter (Single-Sign-on SdK Vetter) to check the logical correctness and identify vulnerabilities of SSO SDKs.

Furthermore, we noticed that token abuse and leakage (ST2) and single point of compromise (ST5) security threats are the most frequently addressed in the literature, with a coverage level of 36.17% and 42.55%, respectively. On the privacy front, unauthorized data collection and sharing (PT1) and user PII leakage (PT5) are the most frequently addressed threats across the analyzed 47 SSO schemes with a coverage level of 40.42% and 51.06%, respectively. For instance, schemes [52,94] propose distributed or threshold-based authentication solutions addressing ST2 and ST5 threats. Additionally, Lux et al. [74] proposed system addresses the risks associated with single points of compromise, token abuse and leakage, and unauthorized data collection and sharing threats by combining decentralized identifiers, verifiable credentials, self-sovereign identity (SSI), and distributed ledger technology. On the other hand, schemes [101,117] effectively mitigated all four threats by employing a range of advanced techniques, including multi-factor authentication, cryptographic methods such as pseudo-random functions, hash functions, signatures, and credentials with secret-sharing thresholds. These approaches were further strengthened by leveraging secret-sharing mechanisms, blockchain technology, smart contracts, the Interplanetary File System (IPFS), decentralized identifiers (DID), and self-sovereign identity solutions.

Moreover, only three of the analyzed SSO schemes proposed solutions that effectively mitigate all the identified privacy threats, which are [17,100], and [112]. EL PASSO [17] is a privacy-preserving, asynchronous SSO system that safeguards users' privacy against both IdPs and RPs through blind issuance, unlinkability, accountability, and non-interactive zero-knowledge properties. It also allows users to select the subsets of attributes to be revealed (e.g., email address but not last name) when signing up or signing on at an RP. The system provides strong privacy guarantees, prevents tracking of user activities, and supports selective attribute disclosure, enhancing user control over their personal information during the authentication process, thus minimizing the risk of PII data leakage and consent manipulation. It also disallows IdPs, RPs, and any colluding sets of these entities from tracking users' access to RPs or correlating accesses by the same user to different sites. Otherwise, García-Rodríguez et al. [100] introduce the initial implementation of the Pointcheval-Sanders Multi-Signatures (PS-MS) cryptographic scheme, which was first proposed in [118]. They also demonstrate its integration into a distributed, privacy-preserving identity management system that was developed as part of the OLYMPUS H2020 European research project. The evaluation proves that the system provides remarkable privacy-preservation features, including unforgeability, minimum disclosure of personal data via zero-knowledge proofs, unlinkability in online transactions, and fully distributed credential issuance. Moreover, MISO [112] introduces a mixer between the user, RP, and IdP, ensuring that IdPs cannot track users' login activities across different websites. The mixer hides the user's login to a particular RP, and it obfuscates user information before forwarding it to the RP. This prevents tracking, profiling, and linking accounts across different services. Additionally, within MISO, users have the ability to disclose only the information they wish to share with RPs. The mixer allows for selective redaction of sensitive attributes, such as email or phone numbers, reducing the risk of PII leakage. To prevent data collection, MISO relies on a Trusted Execution Environment (TEE) to secure the mixer and ensure the confidentiality and integrity of its operations. Further, MISO prevents consent manipulation during the login process since users authenticate through the mixer and their consent is always verified before sharing any data with the RP or IdP.

Furthermore, our systematic literature review (SLR) revealed that none of the selected schemes fully addressed all identified security and privacy threats. However, MISO [112] and our previously proposed scheme [10] stand out by addressing the highest number of these threats. The scheme [10] was designed to efficiently secure the authorization and authentication process and preserve users' privacy of OpenID Connect, the SSO protocol standard. Specifically, our scheme targets ST2, ST5, ST6, PT2, PT4, and PT5 threats by integrating Blockchain technology and non-fungible tokens within the OIDC protocol. On the other hand, MISO [112] addresses, alongside the five privacy threats (i.e., PT1–PT5), the single point of failure threat ST5 by incorporating different IdPs in a single SSO process. MISO allows users to authenticate with a combination of IdPs (e.g., 2 out of 3), ensuring that even if one IdP is compromised or experiences downtime, the user can still log in securely through the others. This feature safeguards against service unavailability and account compromise, enhancing the system's resilience. Despite the strengths of these schemes, it is important to note that they address only six out of the thirteen identified threats, underscoring the need for continued efforts in designing more robust and privacy-preserving solutions for SSO.

## 5 Conclusion and Future Work

This paper presents a systematic literature review providing a comprehensive overview of SSO security and privacy by addressing four critical research questions. Through our analysis, we have identified several key findings and insights that contribute to our understanding of the security and privacy challenges and risks associated with SSO authentication processes, the potential countermeasures to address these challenges, and the state of research on the current SSO schemes. The paper identifies and classifies security and privacy threats that can compromise SSO integrity, availability, and users' privacy. Additionally, it presents the

potential implications of these threats, which span from unauthorized access to sensitive information to the compromise of users' credentials and data breaches, emphasizing the need for robust security and privacy measures. Next, the paper highlights the different countermeasures and mechanisms that can be employed to mitigate the various classified security and privacy threats. This helps researchers and developers make well-informed decisions about the best practices that align with their needs and motivates further research in this area. Finally, we explored and analyzed the state-of-the-art of the current proposed SSO schemes in the literature. This involved assessing the threats each scheme addresses, pinpointing areas where security and privacy enhancements are needed, and identifying opportunities for further research. Our findings reveal that none of the selected schemes have successfully managed to mitigate all the security and privacy threats identified in our SLR. Even worse, the schemes that addressed the highest number of threats only managed to mitigate six out of a total of thirteen.

**Acknowledgement:** The authors acknowledge the Computers, Materials & Continua for their support for the paper.

**Funding Statement:** The authors declare that no funds, grants, or other support were received during the preparation of this work.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Abdelhadi Zineddine and Yousra Belfaik; Data curation, Abdelhadi Zineddine, Yousra Belfaik and Abdeslam Rehaimi; Formal analysis, Abdelhadi Zineddine, Yousra Belfaik and Abdeslam Rehaimi; Investigation, Abdelhadi Zineddine, Yousra Belfaik and Yassine Sadqi; Methodology, Abdelhadi Zineddine, Yousra Belfaik and Yassine Sadqi; Project administration, Yassine Sadqi; Resources, Said Safi; Software, Abdelhadi Zineddine, Yousra Belfaik and Abdeslam Rehaimi; Supervision, Yassine Sadqi and Said Safi; Validation, Yassine Sadqi and Said Safi; Writing—original draft, Abdelhadi Zineddine and Yousra Belfaik; Writing—review and editing, Abdelhadi Zineddine, Yousra Belfaik and Yassine Sadqi. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data supporting the findings of this study are provided in the following GitHub link https://github.com/YOUSRA-BEL/SSO_SLR_SupplementaryDATA.git (accessed on 17 June 2025).

**Ethics Approval:** This declaration is not applicable as the reported research does not involve any data from humans or animals.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Petrosyan A. Internet and social media users in the world 2025; 2025. [cited 2025 Jun 17]. Available from: https://www.statista.com/statistics/617136/digital-population-worldwide/.
2. Gaw S, Felten EW. Password management strategies for online accounts. In: Proceedings of the Second Symposium on Usable Privacy and Security—SOUPS'06. Pittsburgh, PA, USA; 2006. p. 44–55. doi:10.1145/1143120.1143127.
3. Florencio D, Herley C. A large-scale study of web password habits. In: Proceedings of the 16th International Conference on World Wide Web—WWW'07. Banff, AB, Canada; 2007. p. 657–66. doi:10.1145/1242572.1242661.
4. Rehaimi A, Sadqi Y, Maleh Y, Gaba GS, Gurtov A. Towards a federated and hybrid cloud computing environment for sustainable and effective provisioning of cyber security virtual laboratories. Expert Syst Appl. 2024;252(19):124267. doi:10.1016/j.eswa.2024.124267.
5. Liu G, Gao X, Wang H. An investigation of identity-account inconsistency in single sign-on. In: Proceedings of the Web Conference. Ljubljana, Slovenia; 2021. p. 105–17.
6. Morkonda SG, Chiasson S, van Oorschot PC. Empirical analysis and privacy implications in OAuth-based single sign-on systems. In: Proceedings of the 20th Workshop on Privacy in the Electronic Society. New York, NY, USA: Association for Computing Machinery; 2021. p. 195–208. doi:10.1145/3463676.3485600.

7. Baum C, Frederiksen T, Hesse J, Lehmann A, Yanai A. PESTO: proactively secure distributed single sign-on, or how to trust a hacked server. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). Genoa, Italy; 2020. p. 587–606. doi:10.1109/EuroSP48549.2020.00044.
8. Fett D, Küsters R, Schmitz G. A comprehensive formal security analysis of OAuth 2.0. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria; 2016. p. 1204–15.
9. Sadqi Y, Belfaik Y, Safi S. Web OAuth-based SSO systems security. In: Proceedings of the 3rd International Conference on Networking, Information Systems & Security. Marrakech, Morocco; 2020. p. 1–7. doi:10.1145/3386723.3387888.
10. Belfaik Y, Sadqi Y, Maleh Y, Said S, Tawalbeh L, Salah K. A novel secure and privacy-preserving model for OpenID connect based on blockchain. IEEE Access. 2023;11:67660–78. doi:10.1109/access.2023.3292143.
11. Sheik SA, Muniyandi AP. Secure authentication schemes in cloud computing with glimpse of artificial neural networks: a review. Cyber Secur Appl. 2023;1(2):100002. doi:10.1016/j.csa.2022.100002.
12. Hardt D. RFC 6749: the OAuth 2.0 authorization framework; 2012. RFC Editor. [cited 2025 Jun 17]. Available from: https://datatracker.ietf.org/doc/html/rfc6749.
13. Singh J, Chaudhary NK. OAuth 2.0: architectural design augmentation for mitigation of common security vulnerabilities. J Inf Secur Appl. 2022;65(1):103091. doi:10.1016/j.jisa.2021.103091.
14. Sakimura N, Bradley J, Jones M, De Medeiros B, Mortimore C. OpenID connect core 1.0; 2014. The OpenID Foundation. [cited 2025 Jun 17]. Available from: https://openid.net/specs/openid-connect-core-1_0.html.
15. Fett D, Küsters R, Schmitz G. The web SSO standard OpenID connect: in-depth formal security analysis and security guidelines. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). Santa Barbara, CA, USA; 2017. p. 189–202. doi:10.1109/CSF.2017.20.
16. Ghasemisharif M, Ramesh A, Checkoway S, Kanich C, Polakis JO. Where art thou? An empirical analysis of single sign-on account hijacking and session management on the web. In: Proceedings of the 27th USENIX Conference on Security Symposium, SEC'18. Baltimore, MD, USA: USENIX Association; 2018. p. 1475–92.
17. Zhang Z, Król M, Sonnino A, Zhang L, Riviére E. EL PASSO: privacy-preserving, asynchronous single sign-on. arXiv:2002.10289. 2020.
18. Khandelwal S. Uh Oh, Yahoo! data breach may have hit over 1 billion users; 2016. The Hacker News. [cited 2025 Jun 17]. Available from: https://thehackernews.com/2016/09/yahoo-data-breach-billion.html.
19. Kumar M. Facebook admits public data of its 2.2 billion users has been compromised; 2018. The Hacker News. [cited 2025 Jun 17]. Available from: https://thehackernews.com/2018/04/facebook-data-privacy.html.
20. Lakshmanan R. Hackers abused microsoft's "Verified Publisher" oauth apps to breach corporate email accounts; 2023. The Hacker News. [cited 2025 Jun 17]. Available from: https://thehackernews.com/2023/02/hackers-abused-microsofts-verified.html.
21. Radha V, Reddy DH. A survey on single sign-on techniques. Proc Technol. 2012;4:134–9. doi:10.1016/j.protcy.2012.05.019.
22. Cusack B, Ghazizadeh E. Evaluating single sign-on security failure in cloud services. Bus Horiz. 2016;59(6):605–14. doi:10.1016/j.bushor.2016.08.002.
23. Nongbri I, Hadem P, Chettri S. A survey on single sign-on. Int J Creative Res Thoughts. 2018;6(2):595–602. doi:10.5281/ZENODO.5763157.
24. Rathee T, Singh P. A systematic literature mapping on secure identity management using blockchain technology. J King Saud Univ—Comput Inf Sci. 2022;34(8):5782–96. doi:10.1016/j.jksuci.2021.03.005.
25. de Almeida MG, Canedo ED. Authentication and authorization in microservices architecture: a systematic literature review. Appl Sci. 2022;12(6):3023. doi:10.3390/app12063023.
26. Mousavi Z, Islam C, Babar MA, Abuadbba A, Moore K. Detecting misuses of security APIs: a systematic review. arXiv:2306.08869. 2023.
27. Glöckler J, Sedlmeir J, Frank M, Fridgen G. A systematic review of identity and access management requirements in enterprises and potential contributions of self-sovereign identity. Bus Inf Syst Eng. 2023;66(4):421–40. doi:10.1007/s12599-023-00830-x.

28. Kiourtis A, Giannetsos T, Menesidou S, Mavrogiorgou A, Symvoulidis C, Graziani A, et al. Identity management standards: a literature review. Comput Inform. 2023;3(1):35–46.
29. Snyder H. Literature review as a research methodology: an overview and guidelines. J Bus Res. 2019;104(5):333–9. doi:10.1016/j.jbusres.2019.07.039.
30. Kitchenham B, Pearl Brereton O, Budgen D, Turner M, Bailey J, Linkman S. Systematic literature reviews in software engineering—a systematic literature review. Inf Softw Technol. 2009;51(1):7–15. doi:10.1016/j.infsof.2008.09.009.
31. Zaoui M, Yousra B, Yassine S, Yassine M, Karim O. A comprehensive taxonomy of social engineering attacks and defense mechanisms: toward effective mitigation strategies. IEEE Access. 2024;12(2):72224–41. doi:10.1109/access.2024.3403197.
32. Maceiras M, Salehzadeh Niksirat K, Bernard G, Garbinato B, Cherubini M, Humbert M, et al. Know their customers: an empirical study of online account enumeration attacks. ACM Trans the Web. 2024;18(3):1–36. doi:10.1145/3664201.
33. Farooqi S, Zaffar F, Leontiadis N, Shafiq Z. Measuring and mitigating OAuth access token abuse by collusion networks. In: Internet Measurement Conference 2017 (IMC '17). London, UK; 2017. doi:10.1145/3131365.3131404.
34. Lodderstedt T, McGloin M, Hunt P. RFC 6819: OAuth 2.0 threat model and security considerations. USA: RFC Editor; 2013. doi:10.17487/RFC6819.
35. Mainka C, Mladenov V, Schwenk J, Wich T. SoK: single sign-on security—an evaluation of OpenID connect. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P). Paris, France; 2017. p. 251–66. doi:10.1109/EuroSP.2017.32.
36. Guan C, Sun K, Lei L, Wang P, Wang Y, Chen W. DangerNeighbor attack: information leakage via PostMessage mechanism in HTML5. Comput Secur. 2019;80(2):291–305. doi:10.1016/j.cose.2018.09.010.
37. Innocenti T, Golinelli M, Onarlioglu K, Mirheidari A, Crispo B, Kirda E. OAuth 2.0 redirect URI validation falls short, literally. In: Annual Computer Security Applications Conference (ACSAC'23). Austin, TX, USA; 2023, doi:10.1145/3627106.3627140.
38. Li W, Mitchell CJ, Chen T. Mitigating CSRF attacks on OAuth 2.0 systems. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). Belfast, Ireland: IEEE; 2018. p. 1–5.
39. Wei H, Hassanshahi B, Bai G, Krishnan P, Vorobyov K. MoScan: a model-based vulnerability scanner for web single sign-on services. In: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2021). Virtual, Denmark: Association for Computing Machinery; 2021. p. 678–81. doi:10.1145/3460319.3469081.
40. Pandey P, Nisha TN. Challenges in single sign-on. J Phys: Conf Ser. 2021;1964(4):042016. doi:10.1088/1742-6596/1964/4/042016.
41. Li X, Xu J, Zhang Z, Lan X, Wang Y. Modular security analysis of OAuth 2.0 in the three-party setting. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). Genoa, Italy; 2020. p. 276–93. doi:10.1109/EuroSP48549.2020.000256.
42. Yang R, Li G, Lau WC, Zhang K, Hu P. Model-based security testing: an empirical study on OAuth 2.0 implementations. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16). Xi'an, China: Association for Computing Machinery; 2016. p. 651–62. doi:10.1145/2897845.2897874.
43. Arshad E, Benolli M, Crispo B. Practical attacks on login CSRF in OAuth. Comput Secur. 2022;121(4):102859. doi:10.1016/j.cose.2022.102859.
44. Sumongkayothin K, Rachtrachoo P, Yupuech A, Siriporn K. OVERSCAN: OAuth 2.0 scanner for missing parameters. In: Liu J, Huang X, editors. Network and system security. Lecture notes in computer science. Cham, Switzerland: Springer; 2019. Vol. 11928. p. 221–33. doi:10.1007/978-3-030-36938-5_13.
45. Westers M, Wich T, Jannett L, Mladenov V, Mainka C, Mayer A. SSO-monitor: fully-automatic large-scale landscape, security, and privacy analyses of single sign-on in the wild. arXiv:2302.01024. 2023. doi:10.48550/arxiv.2302.01024.

46. Wang C, Xiong Y, Huang W, Xia H, Huang J, Su C. A verified secure protocol model of OAuth dynamic client registration. In: 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM). Chengdu, China; 2017. p. 106–10. doi:10.1109/BIGCOM.2017.50.
47. Zineddine A, Chakir O, Sadqi Y, Maleh Y, Singh Gaba G, Gurtov A, et al. A systematic review of cybersecurity assessment methods for HTTPS. Comput Electr Eng. 2024;115(6):109137. doi:10.1016/j.compeleceng.2024.109137.
48. Li W, Mitchell CJ, Chen T. Your code is my code: exploiting a common weakness in OAuth 2.0 implementations. vol. 26. In: Security Protocols XXVI: 26th International Workshop; 2018 Mar 19–21; Cambridge, UK: Springer. p. 24–41. doi:10.1007/978-3-030-03251-7_3.
49. Navas J, Beltrán M. Understanding and mitigating OpenID connect threats. Comput Secur. 2019;84(10):1–16. doi:10.1016/j.cose.2019.03.003.
50. Qiu K, Liu Q, Liu J, Yu L, Wang Y. An empirical study of OAuth-based SSO system on web. In: Wireless Algorithms, Systems, and Applications: 13th International Conference, WASA 2018; 2018 Jun 20–22; Tianjin, China: Springer; 2018. p. 400–11. doi:10.1007/978-3-319-94268-1_33.
51. Yang R, Lau WC, Chen J, Zhang K. Vetting single sign-on SDK implementations via symbolic reasoning. In: 27th USENIX Security Symposium (USENIX Security 18). Baltimore, MD, USA: USENIX Association; 2018. p. 1459–74.
52. Agrawal S, Miao P, Mohassel P, Mukherjee P. PASTA: password-based threshold authentication. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. Toronto, ON, Canada; 2018. p. 2042–59. doi:10.1145/3243734.3243839.
53. Zineddine A, Belfaik Y, Sadqi Y. A deep dive into cybersecurity risk assessment and countermeasures in online social networks. In: Risk assessment and countermeasures for cybersecurity. Hershey, PA, USA: IGI Global; 2024. p. 1–19. doi:10.4018/979-8-3693-2691-6.ch001.
54. Bilal M, Wang C, Yu Z, Bashir A. Evaluation of secure OpenID-based RAAA user authentication protocol for preventing specific web attacks in web apps. In: 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS). Beijing, China; 2020. p. 82–90. doi:10.1109/ICSESS49938.2020.9237635.
55. Belfaik Y, Lmouhandiz A, Sadqi Y, Safi S. Single sign-on revocation access. In: Maleh Y, Alazab M, Gherabi N, Tawalbeh L, Abd El-Latif AA, editors. Advances in information, communication and cybersecurity. ICI2C 21. Lecture notes in networks and systems. Cham, Switzerland: Springer; 2022. Vol. 357, p. 535–44. doi:10.1007/978-3-030-91738-8_49.
56. Sudhodanan A, Paverd A. Pre-hijacked accounts: an empirical study of security failures in user account creation on the web. arXiv:2205.10174. 2022. doi:10.48550/ARXIV.2205.10174.
57. Ghasemisharif M, Kanich C, Polakis J. Towards automated auditing for account and session management flaws in single sign-on deployments. In: 2022 IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA: IEEE; 2022. doi:10.1109/sp46214.2022.9833753.
58. Zineddine A, Sadqi Y. Understanding the digital frontier: examining privacy and data security in desktop web browsers. In: Motahhir S, Bossoufi B, editors. Digital technologies and applications. ICDTA 2024. Lecture notes in networks and systems. Cham, Switzerland: Springer; 2024. Vol. 1098, p. 138–47. doi:10.1007/978-3-031-68650-4_14.
59. Drakonakis K, Ioannidis S, Polakis J. The cookie hunter: automated black-box auditing for web authentication and authorization flaws. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS'20). Virtual Event, USA: Association for Computing Machinery; 2020. p. 1953–70. doi:10.1145/3372297.3417869.
60. Gao X, Yu L, He H, Wang X, Wang Y. A research of security in website account binding. J Inf Secur Appl. 2020;51(1):102444. doi:10.1016/j.jisa.2019.102444.
61. Beer Mohamed MI, Hassan MF, Safdar S, Saleem MQ. Adaptive security architectural model for protecting identity federation in service oriented computing. J King Saud Univ—Comput Inf Sci. 2021;33(5):580–92. doi:10.1016/j.jksuci.2019.03.004.
62. Bao X, Zhang X, Lin J, Chu D, Wang Q, Li F. Towards the trust-enhancements of single sign-on services. In: 2019 IEEE Conference on Dependable and Secure Computing (DSC). Hangzhou, China; 2019. p. 1–8. doi:10.1109/DSC47296.2019.8937676.

63. Chu D, Lin J, Li F, Zhang X, Wang Q, Liu G. Ticket transparency: accountable single sign-on with privacy-preserving public logs. In: International Conference on Security and Privacy in Communication Systems. Orlando, FL, USA: Springer; 2019. p. 511–31.
64. Jannett L, Mladenov V, Mainka C, Schwenk J. DISTINCT: identity theft using in-browser communications in dual-window single sign-on. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS'22). Los Angeles, CA, USA: Association for Computing Machinery; 2022. p. 1553–67. doi:10.1145/3548606.3560692.
65. Ramamoorthi LS, Sarkar D. Single sign-on: a solution approach to address inefficiencies during sign-out process. IEEE Access. 2020;8:195675–91. doi:10.1109/ACCESS.2020.3033570.
66. Ramamoorthi L, Sarkar D. Single sign-on implementation: leveraging browser storage for handling tabbed browsing sign-outs. In: Developments and Advances in Defense and Security: Proceedings of MICRADS 2019. Singapore: Springer; 2020. p. 15–28.
67. Sassetti G, Sharif A, Sciarretta G, Carbone R, Ranise S. Assurance, consent and access control for privacy-aware OIDC deployments. In: Atluri V, Ferrara AL, editors. Data and applications security and privacy XXXVII. DBSec 2023. Lecture notes in computer science. Cham, Switzerland: Springer; 2023. Vol. 13942, p. 203–22. doi:10.13140/RG.2.2.10724.35202.
68. Friebe S, Sobik I, Zitterbart M. DecentID: decentralized and privacy-preserving identity storage system using smart contracts. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). New York, NY, USA: IEEE; 2018. doi:10.1109/trustcom/bigdatase.2018.00016.
69. Mir O, Roland M, Mayrhofer R, Meghias D. Decentralized, privacy-preserving, single sign-on. Secur Commun Netw. 2022;2022(4):1–18. doi:10.1155/2022/9983995.
70. Alaca F, Oorschot PCV. Comparative analysis and framework evaluating web single sign-on systems. ACM Comput Surv. 2020;53(5):112. doi:10.1145/3409452.
71. Belfaik Y, Zineddine A, Sadqi Y, Safi S. Privacy-preserving techniques for online social networks data. In: Risk assessment and countermeasures for cybersecurity. IGI Global; 2024. p. 62–78. doi:10.4018/979-8-3693-2691-6.ch004.
72. Saito T, Shibata S, Kikuta T. Comparison of OAuth/OpenID connect security in America and Japan. In: Barolli L, Li KF, Enokido T, Takizawa M, editors. Advances in networked-based information systems. Cham: Springer International Publishing; 2021. p. 200–10. doi:10.1007/978-3-030-57811-4_19.
73. Kihara M, Iriyama S. Security and performance of single sign-on based on one-time pad algorithm. Cryptography. 2020;4(2):16. doi:10.3390/cryptography4020016.
74. Lux ZA, Thatmann D, Zickau S, Beierle F. Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials. In: 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS). Paris, France: IEEE; 2020. p. 71–8.
75. Karegar F, Gerber N, Volkamer M, Fischer-Hübner S. Helping john to make informed decisions on using social login. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC'18). Pau, France: Association for Computing Machinery; 2018. p. 1165–74. doi:10.1145/3167132.3167259.
76. Hammann S, Sasse R, Basin D. Privacy-Preserving OpenID Connect. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS'20). Taipei, Taiwan: Association for Computing Machinery; 2020. p. 277–89. doi:10.1145/3320269.3384724.
77. Benolli M, Mirheidari SA, Arshad E, Crispo B. The full gamut of an attack: an empirical analysis of OAuth CSRF in the wild. In: Bilge L, Cavallaro L, Pellegrino G, Neves N, editors. Detection of intrusions and malware, and vulnerability assessment. DIMVA 2021. Lecture notes in computer science. Vol. 12756. Cham, Switzerland: Springer; 2021. p. 21–41. doi:10.1007/978-3-030-80825-9_2.
78. Li W, Mitchell CJ. User access privacy in OAuth 2.0 and OpenID connect. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Genoa, Italy: IEEE; 2020. doi:10.1109/eurospw51379.2020.00095.
79. Kuo WC, Shih PW, Huang YC, Wuu LC. An anonymous and authentication protocol for multi-server. Inf Technol Control. 2017;46(2):235–45. doi:10.5755/j01.itc.46.2.13781.

80. Han J, Chen L, Schneider S, Treharne H, Wesemeyer S, Wilson N. Anonymous single sign-on with proxy re-verification. IEEE Trans Inf Forensics Secur. 2020;15:223–36. doi:10.1109/TIFS.2019.2919926.
81. Han J, Chen L, Schneider S, Treharne H, Wesemeyer S. Anonymous single-sign-on for n designated services with traceability. In: Computer Security: 23rd European Symposium on Research in Computer Security, ESORICS 2018; 2018 Sep 3–7; Barcelona, Spain: Springer; 2018. p. 470–90.
82. Wang K, Bai G, Dong N, Dong JS. A framework for formal analysis of privacy on SSO protocols. In: Barolli L, Li KF, Enokido T, Takizawa M, editor. Security and Privacy in Communication Networks: 13th International Conference, SecureComm 2017. Springer; 2018. p. 763–77 doi:10.1007/978-3-319-78813-5_41.
83. Morkonda SG, Chiasson S, van Oorschot PC. "Sign in with. Privacy": timely disclosure of privacy differences among web SSO login options. ACM Trans Priv Secur. 2025;28(2):1–28. doi:10.1145/3711898.
84. Ramamoorthi L, Sarkar D. Single sign-on demystified: security considerations for developers and users. In: Rocha Á, Adeli H, Reis L, Costanzo S, editors. Trends and advances in information systems and technologies. WorldCIST'18 2018. Advances in intelligent systems and computing. Vol. 746. Cham, Switzerland: Springer; 2018. p. 185–96. doi: 10.1007/978-3-319-77712-2_18.
85. Mozilla. Window: postMessage() method—Web APIs—MDN—developer.mozilla.org. [cited 2025 Jun 17]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage.
86. Dashti S, Sharif A, Carbone R, Ranise S. Automated risk assessment and what-if analysis of OpenID connect and OAuth 2.0 deployments. In: Barker K, Ghazinour K, editors. Data and applications security and privacy XXXV. DBSec 2021. Lecture notes in computer science. Vol. 12840, Cham, Switzerland: Springer; 2021. p. 325–37. doi:10.1007/978-3-030-81242-3_19.
87. Mozilla. Intersection Observer API—Web APIs—MDN—developer.mozilla.org; [cited 2025 Jun 17]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/Intersection_Observer_API.
88. Belfaik Y, Lotfi Y, Sadqi Y, Safi S. A comparative study of protocols' security verification tools: Avispa, scyther, ProVerif, and Tamarin. In: Motahhir S, Bossoufi B, editors. Digital technologies and applications. ICDTA 2024. Lecture notes in networks and systems. Cham, Switzerland: Springer; Vol. 1099, 2024. p. 118–28. doi:10.1007/978-3-031-68653-5_12..
89. Veronese L, Calzavara S, Compagna L. Bulwark: holistic and verified security monitoring of web protocols. In: Chen L, Li N, Liang K, Schneider S, editors. Computer security–ESORICS 2020. Lecture notes in computer science. Vol. 12308. Cham, Switzerland: Springer; 2020. p. 23–41. doi:10.1007/978-3-030-58951-6_2.
90. Bilal M, Asif M, Bashir A. Assessment of secure OpenID-based DAAA protocol for avoiding session hijacking in web applications. Secur Commun Netw. 2018;2018(11):1–10. doi:10.1155/2018/6315039.
91. Morkonda SG, Chiasson S, van Oorschot PC. Influences of displaying permission-related information on web single sign-on login decisions. Comput Secur. 2024;139(5):103666. doi:10.1016/j.cose.2023.103666.
92. Rehaimi A, Sadqi Y, Maleh Y. A comparative study of online cybersecurity training platforms. In: Ben Hedia B, Maleh Y, Krichen M, editors. Verification and evaluation of computer and communication systems VECoS 2023. Lecture notes in computer science. Vol. 14368. Cham, Switzerland: Springer; 2024. p. 122–34. doi:10.1007/978-3-031-49737-7_9.
93. Argyriou M, Dragoni N, Spognardi A. Security flows in OAuth 2.0 framework: a case study. In: Tonetta S, Schoitsch E, Bitsch F, editors. Computer safety, reliability, and security. Cham: Springer International Publishing; 2017. p. 396–406.
94. Magnanini F, Ferretti L, Colajanni M. Flexible and survivable single sign-on. In: Cyberspace Safety and Security: 13th International Symposium, CSS 2021. Vol. 13. Cham: Springer; 2022. p. 182–97. doi:10.1007/978-3-030-94029-4_13.
95. Philippaerts P, Preuveneers D, Joosen W. OAuch: exploring security compliance in the OAuth 2.0 ecosystem. In: Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID'22). Limassol, Cyprus: ACM; 2022. p. 460–81. doi:10.1145/3545948.3545955.
96. Rahat TA, Feng Y, Tian Y. Cerberus: query-driven scalable vulnerability detection in oauth service provider implementations. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications

Security. CCS'22. New York, NY, USA: Association for Computing Machinery; 2022. p. 2459–73. doi:10.1145/3548606.3559381.
97. Deeptha R, Mukesh R. Extending OpenID connect towards mission critical applications. Cybern Inf Technol. 2018;18(3):93–110. doi:10.2478/cait-2018-0041.
98. Villarán C, Beltrán M. User-centric privacy for identity federations based on a recommendation system. Electronics. 2022;11(8):1238. doi:10.3390/electronics11081238.
99. Bilal M, Showngwe C, Bashir A, Ghadi Y. Assessing secure OpenID-Based EAAA protocol to prevent MITM and phishing attacks in web apps. Comput Mater Contin. 2023;75(3):4713–33. doi:10.32604/cmc.2023.037071.
100. García-Rodríguez J, Torres Moreno R, Bernal Bernabe J, Skarmeta A. Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures. J Inf Secur Appl. 2021;62(2):102971. doi:10.1016/j.jisa.2021.102971.
101. Das D, Sethuraman SC, Satapathy SC. A decentralized open web cryptographic standard. Comput Electr Eng. 2022;99(19):107751. doi:10.1016/j.compeleceng.2022.107751.
102. Cha SC, Chang CL, Xiang Y, Huang TJ, Yeh KH. Enhancing OAuth with blockchain technologies for data portability. IEEE Trans Cloud Comput. 2023;11(1):349–66. doi:10.1109/TCC.2021.3094846.
103. Guo C, Lang F, Wang Q, Lin J. UP-SSO: enhancing the user privacy of SSO by integrating PPID and SGX. In: 2021 International Conference on Advanced Computing and Endogenous Security. Nanjing, China: IEEE; 2022. doi:10.1109/ieeeconf52377.2022.10013340.
104. Zhang Z, Xu C, Jiang C, Chen K. TSAPP: threshold single-sign-on authentication preserving privacy. IEEE Trans Dependable Secure Comput. 2023;21(4):1–13. doi:10.1109/tdsc.2023.3285393.
105. Patel S, Sahoo A, Mohanta BK, Panda SS, Jena D. DAuth: a decentralized web authentication system using ethereum based blockchain. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). Vellore, India: IEEE; 2019. doi:10.1109/vitecon.2019.8899393.
106. Asghar MR, Backes M, Simeonovski M. PRIMA: privacy-preserving identity and access management at internet-scale. In: 2018 IEEE International Conference on Communications (ICC). Kansas City, MO, USA; 2018. p. 1–6. doi:10.1109/ICC.2018.8422732.
107. Reddy GS, Konala DTR. Easeid- a session-based single sign-on self-sovereign identity and access management system using blockchain. Indian J Comput Sci Eng. 2022;13(4):1197–209. doi:10.21817/indjcse/2022/v13i4/221304176.
108. Diaz Rivera JJ, Akbar W, Ahmed Khan T, Muhammad A, Song WC. Secure enrollment token delivery mechanism for zero trust networks using blockchain. IEICE Trans Commun. 2023;E106.B(12):1293–301. doi:10.1587/transcom.2022tmp0005.
109. Jiang J, Wang D, Zhang G, Chen Z. Quantum-resistant password-based threshold single-sign-on authentication with updatable server private key. In: Atluri V, Di Pietro R, Jensen CD, Meng W, editors. Computer security—ESORICS–2022. Lecture notes in computer science. Cham, Switzerland: Springer; Vol. 13555, 2022. p. 295–316. doi:10.1007/978-3-031-17146-8_15.
110. Rushdy E, Khedr W, Salah N. Framework to secure the OAuth 2.0 and JSON web token for REST API. J Theor Appl Inf Technol. 2021;99(9): 2144–61.
111. Liu S, Song Q,Sun K, Li Q. SGX-cube: an SGX-enhanced single sign-on system against server-side credential leakage. In: Park N, Sun K, Foresti S, Butler K, Saxena N, editors. Security and privacy in communication networks. SecureComm 2020. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Cham, Switzerland: Springer; 2020. Vol. 336, p. 275–90. doi: 10.1007/978-3-030-63095-9.
112. Xu R, Yang S, Zhang F, Fang Z. MISO: legacy-compatible privacy-preserving single sign-on using trusted execution environments. In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P). Delft, Netherlands: IEEE; 2023. p. 352–72. doi:10.1109/EuroSP57164.2023.00029.
113. Johnson AD, Alom I, Xiao Y. Rethinking single sign-on: a reliable and privacy-preserving alternative with verifiable credentials. In: Proceedings of the 10th ACM Workshop on Moving Target Defense. Copenhagen, Denmark; 2023. p. 25–8. doi:10.1145/3605760.3623767.

114. Mahnamfar A, Bicakci K, Uzunay Y. ROSTAM: a passwordless web single sign-on solution mitigating server breaches and integrating credential manager and federated identity systems. Comput Secur. 2024;139(5):103739. doi:10.1016/j.cose.2024.103739.
115. Kalantari S, Philippaerts P, Dimova Y, Hughes D, Joosen W, De Decker B. A user-centric approach to API delegations: . In: Tsudik G, Conti M, Liang K, Smaragdakis G, editors. Computer security–ESORICS 2023. Lecture notes in computer science. Cham, Switzerland: Springer; Vol. 14345, 2023. p. 318–37. doi:10.1007/978-3-031-51476-0_16.
116. Frederiksen TK, Hesse J, Poettering B, Towa P. Attribute-based Single sign-on: secure, private, and efficient; 2023. Cryptology ePrint Archive. [cited 2025 Jun 17]. Available from: https://eprint.iacr.org/2023/915.
117. Krishna DP, Ramaguru R, Praveen K, Sethumadhavan M, Ravichandran KS, Krishankumar R, et al. SSH-DAuth: secret sharing based decentralized oauth using decentralized identifier. Sci Rep. 2023;13(1):18335. doi:10.1038/s41598-023-44586-6.
118. Camenisch J, Drijvers M, Lehmann A, Neven G, Towa P. Short threshold dynamic group signatures. In: Galdi C, Kolesnikov V, editors. Security and cryptography for networks. SCN 2020. Lecture notes in computer science. Vol. 12238. Cham, Switzerland: Springer; 2020. p. 401–23. doi:10.1007/978-3-030-57990-6_20.