Privacy-Preserving Identity Management in Cloud Environments via SOLID

Alfredo Cuzzocrea[1,2,*], Islam Belmerabet[1] and Ismail Benlaredj[1]

[1] *iDEA Lab, University of Calabria, Rende, Italy*
[2] *Dept. of Computer Science, University of Paris City, Paris, France*

Abstract

Focusing on the main research context of *privacy-preserving identity management in multi-Cloud environments*, this paper presents a novel framework integrating high-level identity verification policies, *Zero-Knowledge Proof* (ZKP) protocols, and SOLID *decentralized secure data stores.* By leveraging SOLID *Personal Online Datastores* (PODs), our approach ensures that users maintain full control over their identity data while enabling secure interoperability across diverse Cloud infrastructures. To support our proposed methodology, we provide UML-based modeling for conceptual representation blueprint outlining its implementation.

Keywords

Privacy-Preserving Identity Management, Advanced Identity Management Protocols, SOLID, Multi-Cloud Environments

1. Introduction

Nowadays, *security challenges* faced by *Cloud Computing* (e.g., [13,20]) and the ever-evolving cybercrime landscape present fresh risks that necessitate the adoption of solutions capable of helping companies with prevention, detection, and response. During the previous year, a noteworthy 80% of organizations encountered at least one *Cloud security threat*, while 45% reported experiencing up to four incidents over the same duration. This has given birth to several research proposals in the actual literature.

On the other hand, *Identity Management* (IdM) is a key aspect of *security* and *privacy* of complex information systems (e.g., [19]). When applied to Cloud Computing Environments, the problem gets worse, since security mechanisms are critical for *Cloud services* (e.g., [13,20]). *Data leakage* are a first reason of security issues in Clouds, and it is becoming one of most important challenges of next-generation research. Indeed, nowadays, Clouds influence a wide spectrum of architecture paradigms, including *Mobile Cloud Computing* (MCC) and *Service-Oriented Architectures* (SOA). Data leakage can involve in *theft* of *sensitive personal information* (e.g., [6]), which would seriously limit the large adoption of Cloud Computing technologies in modern *data-driven societies*

For ensuring Cloud security over sensitive information, the actual approaches adopt the idea of employing a *third-party authority service* (e.g., [1]), which ensures about the identity of the actual user. Generally, this is achieved via *Identity and Access Management* (IAM), whose main solution consists in keeping a *single ID* associated to the actual user, and incorporating suitable *identity provisioning* and *identity de-provisioning* activities across different and possibly heterogenous networks (e.g., [21]). As a consequence, it clearly follows that identity management is a relevant research challenge of Cloud Computing security research, especially when it is considered in the context of *multiple Cloud environments*, as also pinpointed by recent studies in the field (e.g., [16,17]). Beyond this, another research challenge arises. How to perform secure identity management in multi-Cloud environments without violating the *privacy* of user data (e.g., [3])? The latter question

[1] *SEDB 2025: 33rd Symposium on Advanced Database Systems, June 16-19, 2025, Ischia, Italy*

* This research has been made in the context of the Excellence Chair in Big Data Management and Analytics at University of Paris City, Paris, France

* *Corresponding author.*

: alfredo.cuzzocrea@unical.it (A. Cuzzocrea); islam.belmerabet@unical.it (I. Belmerabet); ismail.benlaredj@unical.it (I. Benlaredj)

: 0000-0002-7104-6415 (A. Cuzzocrea); 0009-0003-7878-0991 (I. Belmerabet); 0009-0003-1138-8039 (I. Benlaredj)

has become crucial, recently, by also getting a great deal of attention from both the academic and industrial research communities (e.g., [4,5]).

SOLID [18] is a *formal specification* of a network protocols that enables individuals and groups to *securely* store their data in decentralized data stores known as PODs (*Personal Online Datastores*). PODs serve as *secure web servers for data*. By storing data in a POD, the owners retain control over who can access it, including both individuals and applications. A SOLID POD stores and retrieves all data using widely adopted and compatible data formats and protocols, thus being strongly oriented to *architecture interoperability*. To this end, SOLID employs a unified and widely understood approach to describing entities and their interconnectedness, enabling diverse applications to seamlessly collaborate with shared data. This exceptional capability of SOLID allows *multiple applications* to operate on the same dataset, being the latter stored in so-called *SOLID Decentralized Secure Data Stores*.

From the considerations above, it is obvious that, by integrating SOLID in suitable *Cloud services*, privacy-preserving identity management over multiple Clouds can become a reality, thus determining novel, advanced *Cloud-based Big Data applications* (e.g., [3,4,5,25]) where the identity management task is performed in a privacy-preserving manner thanks to SOLID PODs that store user data across these multiple Clouds. On the basis of this conceptual and theoretical setting, in this paper we introduce a framework that makes use of (*i*) *high-level identity verification policies*, (*ii*) *Zero-Knowledge proof* protocols [9], and (*iii*) SOLID to finally obtain privacy-preserving identity management in multiple Cloud environments.

To address security challenges, Cloud-based IAM solutions are designed to provide seamless access control, authentication, and identity verification for IoT devices and users. These solutions enable enterprises to manage *user roles*, enforce *security policies*, and *monitor access permissions* across a distributed IoT ecosystem. *High-risk devices* such as *smart security cameras*, *industrial sensors*, and *connected medical equipment* require advanced IAM frameworks to prevent unauthorized access.

This paper significantly extends our previous study [14], where we introduce the basic concepts and ideas of our proposed framework. The remaining part of this paper is organized as follows. Section 2 introduces the UML modelling of the proposed methodology. Finally, Section 3 reports conclusions and future work of our research.

2. Identity Management Methodology: UML Modelling

We introduce an identity management methodology known as anonymous identification that focuses on entities and identities. This approach utilizes Zero-knowledge proof to authenticate entities without revealing their identifiers. Figure 1 [14] below illustrates the concept of anonymous identification and the IdM service topology within the context of our reference Cloud framework, which includes SOLID PODs to enhance the privacy-preserving capabilities.

One of the key advantages of anonymous identity is the ability to authenticate claims or assertions without requiring any credentials. Let's consider a scenario: a customer makes a book purchase from Amazon and needs to provide their mailing address for delivery. In certain cases, different parties involved in the transaction may require specific information from the user. The shipping company requires the address information. On the other hand, Amazon does not require the customer's address, but it does aim to ensure that the user provides a valid address for the delivery service. To achieve this, following anonymous identification, the IdM service generates a token that includes the necessary address details. In addition to the address, this token contains metadata, access control restrictions, and VM. The token is then transmitted to the CSP, which can subsequently distribute it to the mailing company. The IdM service ensures the secure transmission of user attributes to the CSP, allowing us to utilize these attributes on untrusted hosts and send tokens.

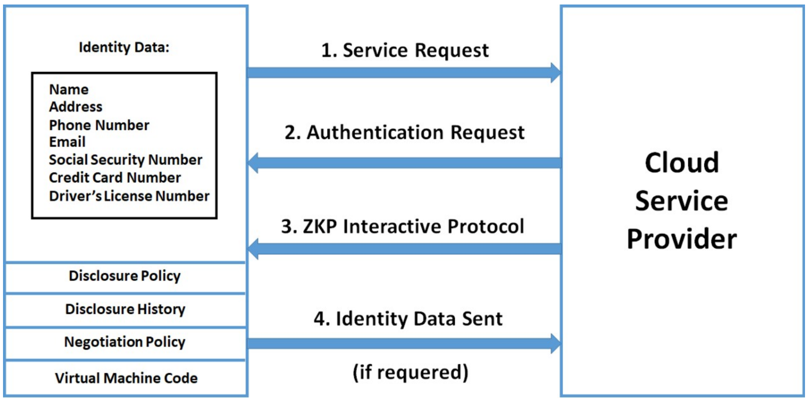

Figure 1: IdM Service Model [14].

2.1. Identity Provider

User identities are stored and disclosed under the control of an IdM service. As shown in Figure 1, it has the following structure: (i) Identity Data: The data utilized for authentication, service acquisition, and service use (for example, SSN and date of birth). This information is encrypted and contained within the IdM service; (ii) Disclosure Policy: This is a set of criteria for selecting identity data from a collection of identities in the IdM service. For example, if particular identification data has been used for a specific service, the same data must be utilized and disclosed every time for the same service. There is no need to reveal any more user identification attributes to that service; (iii) Disclosure History: This may be used for logging and auditing, as well as choosing which Identity data to publish depending on past disclosures; (iv) Negotiation Policy: Based on Zero-Knowledge Proofing, this is Anonymous Identification. It is discussed in the Section 2.2; (v) Virtual Machine: Contains the code for securing user attribute data on untrusted hosts. It is in charge of enforcing the disclosure policies.

2.2. A Service-Oriented Identity Management Framework

According to Web Service Oriented Architecture (WSOA) [15], the IdM framework complexity is contained at a set of service interfaces that should not include business domain-specific features. The basic purpose of the IdM framework is to enable access control to validate authorization for service consumption at runtime. The first condition for access control is an authentication mechanism that checks any possible credentials. This can be done once with validity for a series of future accesses (relates to a single sign-on method) or on every access - which is not preferred in WSOA due to the large number of services that are generally accessed. User authentication can begin, for example, at the WSOA portal layer. Secondly, in order for the authenticated subject to call a WSOA service, an authorization verification mechanism that verifies if permission has been given is required. For both basic and composite services, the functionality of both (authorization verification and authentication) should be wrapped at service interfaces with “identity as a service” features. This suggests that they only provide the IdM services with pertinent data to calculate access control.

The layout of our Web service-aware IdM framework is shown in Figure 2. Three categories of aspects are of interest: IdM services related to the administration and core concern portion of WSOA are the first. The data that the components work on comes in last but not least, followed by the service that implements the components. We utilize this structure to explain our identity management architectural design in the following.

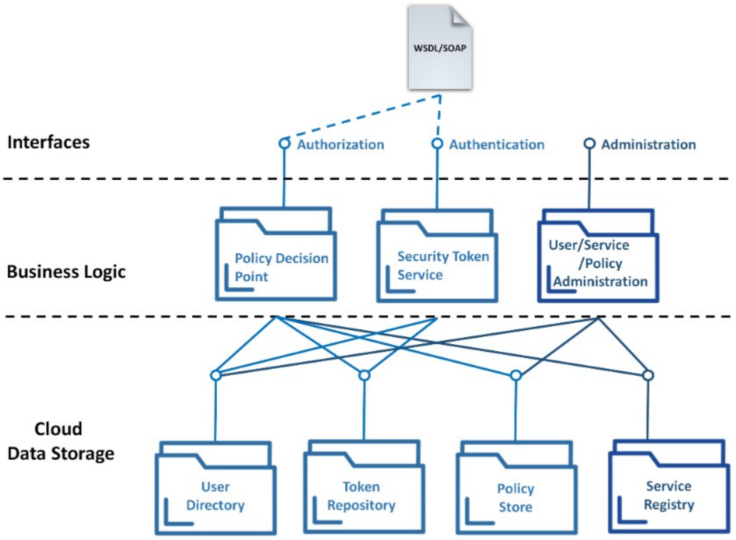

Figure 2: Blueprint of a WSOA-Aware IdM Framework.

Access control is founded on authentication and implies permission verification, both of which are often different procedures. Nowadays, access control is often managed within an application system boundary. The usual application boundaries are ignored with WSOA. Instead, Web services are handled, revealing the essential concerns of apps. They import all functionality required for access control utilizing external service invocations, following the principles of “identity as a service”.

Authentication is handled via the individual Web service interface, which provides several functions to validate various forms of credentials such as username/password-based authentication, certificate-based authentication, and so on. On successful authentication, a security token (creating a session context) with WSOA-wide validity and the option of time-limitation is provided to enable single sign-on and improve privacy. Before accessing protected Web services, user authentication can be initiated at the WSOA portal layer.

The foundation of authorization verification is an access control paradigm. In order to improve on the work of [24] and [8], we introduce an access control meta-model for Web service-oriented framework, like in [7]. In summary, because Web services are specified at a high granularity, it is important to know which user is attempting to access which Web service activity and what the parameters of the submitted invocation are for access control. Every Web service activity is given a unique identifier for identification purposes. A Web service action that requires access control sends its identifier, the user security token, and the parameter the user sent over to the authorization verification service. The permission verification service determines a Boolean result based on its internal policy data, returns it, and instructs the Web service to either continue or cease activities.

Our IdM framework has an administration interface as its third interface. As will be explained later, it serves to maintain the data. It does not always mean WSDL/SOAP because administration is frequently carried out by people.

Using UML 2.0 sequence diagrams, we show the authentication process and the parties involved on the left side of Figure 3. The (virtual) boundary between the IdM framework and the main area of concern for WSOA is where the gap is between the Security Token Service and the Portal on the left, and the Policy Enforcement Point and the Policy Decision Point on the right. Single sign-on capabilities of the WSOA are made possible by issuing security tokens and creating a session context in the process.

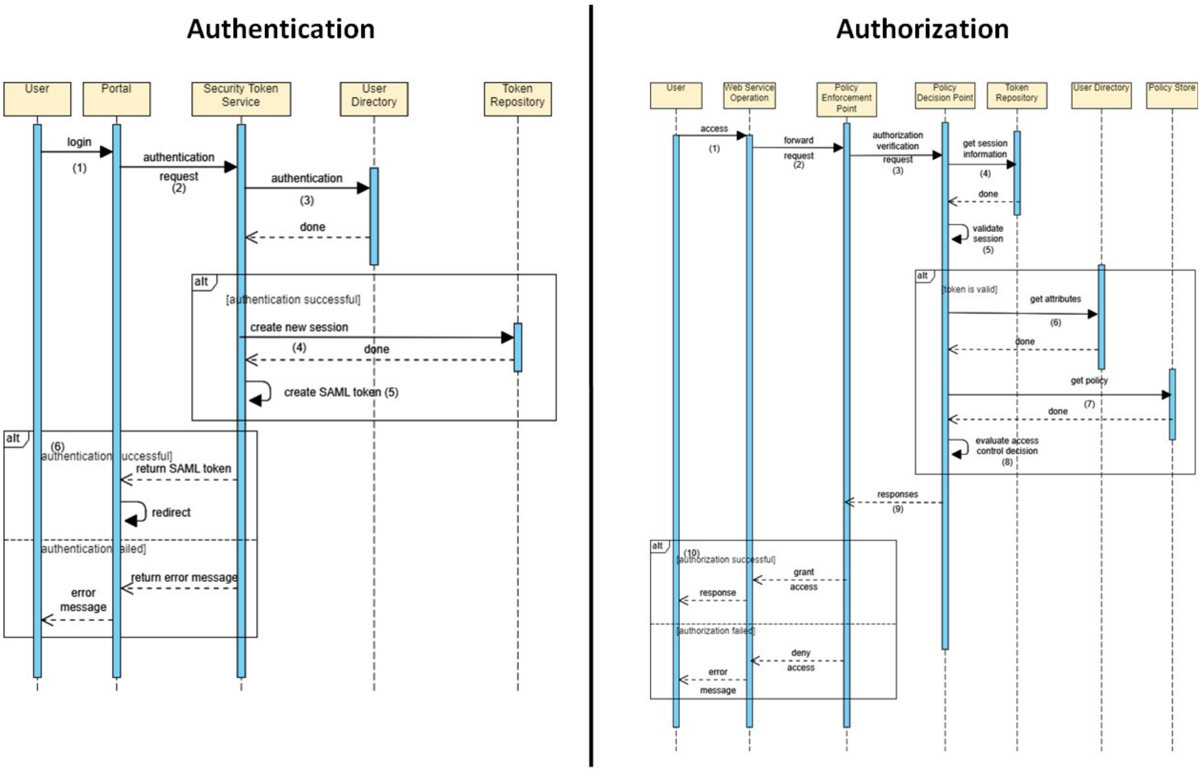

Figure 3: Authentication and Authorization Verification Processes.

The authorization verification procedure is shown on the right-hand side. A Secure Service Agent called the Policy Enforcement Point is installed once on each application server.

2.3. Modelling Digital Identities

According to [12], an identity is a representation of an entity in a particular application domain. The practice of representing and identifying entities as digital identities in virtual networks is known as identity management [11].

In the real world, an entity might be a person, an organization, or a smart gadget. In various settings, an entity may assume different identities. For instance, a student may simultaneously have an account with the teaching system and a bank account. A portion of each account private data.

A collection of an entity attributes, including name, gender, address, and production sequence number of devices, make up an identity. When an attribute is used for authentication, it can be referred to as a credential. Credentials may include a password, a USB drive that contains a special private key, or a fingerprint. Multiple credentials can be associated with one identity. For instance, a user fingerprint or password can be used to access their computer. Every identity has a unique identifier that serves as a means of unambiguous identity identification in the given situation.

A system in charge of establishing, preserving, and administering identities is known as an identity provider (IdP). A Cloud-based system that offers services to consumers is called a Cloud service provider. Although CSPa often play the role of IdP in current identity management practices, IdP may occasionally function independently of CSPs.

Figure 4 shows the connections between entities, identities, identifiers, and credentials/attributes. We present a generic information model of identity, which consists of an identifier set, credential set, attribute set, access information set, and authorization information set, to suit the requirements of identity management. In the following, we provide a description about all the mentioned components, by highlighting their specific characteristics.

- *Identifier Set*: Global IdP generates the global identifiers. The following parameters are used by local IdP to construct the local identifier: local identifiers, global identifiers, and local IdP identifiers. Local IdPs may occasionally provide temporary IDs for brief usage;

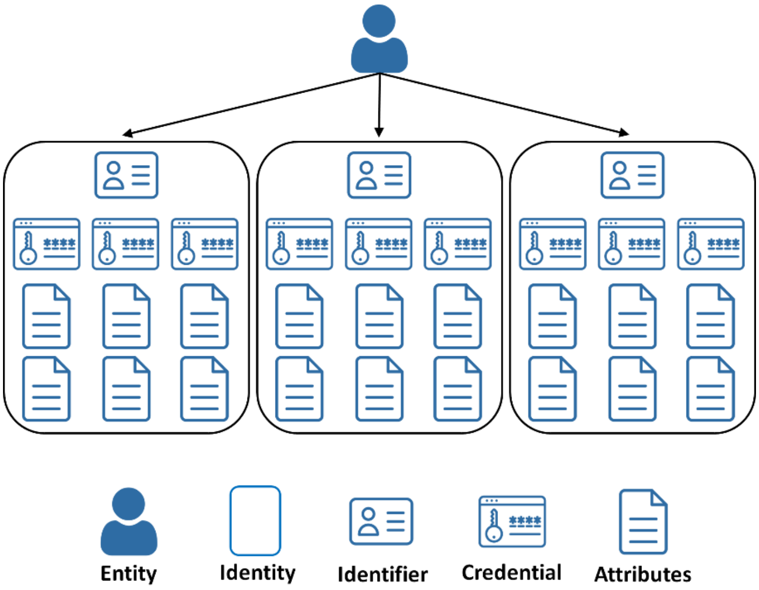

Figure 4: Relationships between Entity, Identity and Credentials/Attributes.

- *Credential Set*: An identity provider attestation of an entity identity, access, or credit is known as a credential. There are several sorts of credentials that an entity may possess, such as local serial numbers, names, values, security levels, descriptions, and values of these credentials;

- *Attribute Set*: The attributes of an attribute set describe the particular context of an entity. This comprises: local attribute serial numbers, attribute types, attribute names, attribute values, and, if these are important attributes, attribute descriptions;

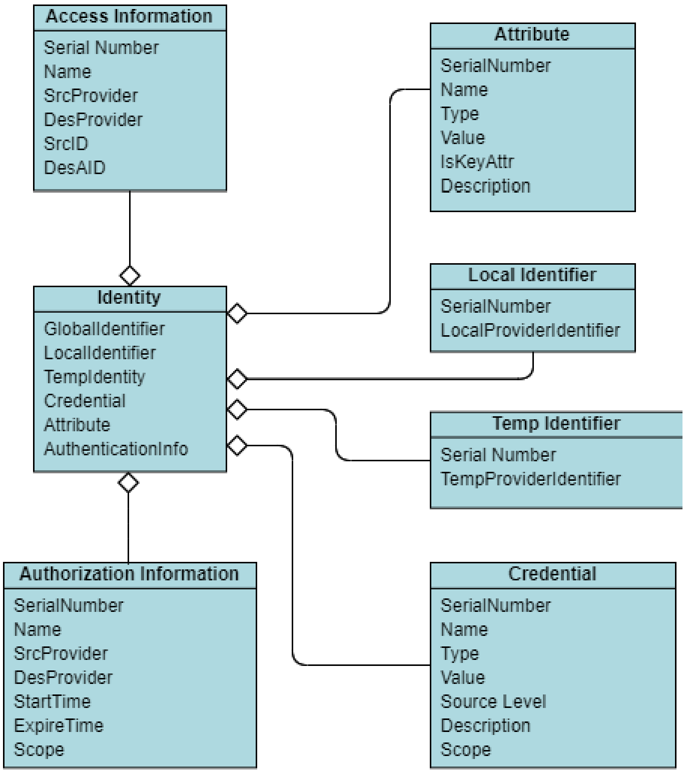

Figure 5: Information Model of Digital Identity.

- *Access Information Set*: The entity cross-domain access is recorded using the access information set. Included in the access information are the following: local access information serial numbers, access source and destination domain identifiers, local entity identifiers in source and destination domains, and permission data;

- *Authorization Information Set*: The authorization between an identity provider and a service provider, or between many service providers, is recorded in the authorization information set. Authorization information consists of local serial numbers for authorization data, identifiers for the source and destination domains, start and expiration periods for authorization, and authorization scopes.

Figure 5 shows the UML class diagram about the specific interaction between IdP and SOLID for managing digital identities, by stressing their cooperation in finally supporting the IdM process across multi-Clouds.

2.4. IdP Interaction with SOLID

Enterprise apps are often installed and operated on the corporate network. Many of these programs are designed to interface with corporate directories, such Microsoft Active Directory, in order to retrieve user profile and group information. More significantly, the directory is usually used to store and verify user credentials. For instance, her/his sign-in credentials are his Active Directory credentials if she/he utilizes SharePoint and Exchange that are hosted on-premises as shown in Figure 6.

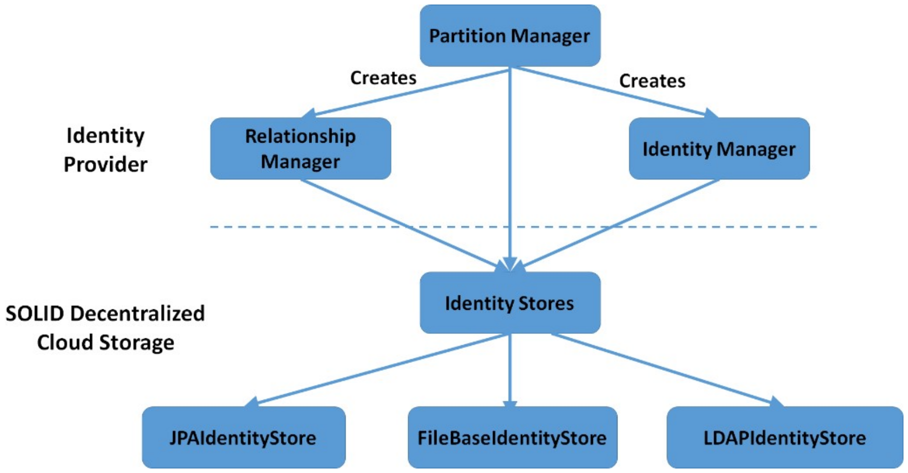

Figure 6: IdP Interaction with SOLID.

However, many applications have migrated outside of a company domain due to growing cooperation and the shift towards Cloud-based settings. Federated Authentication offers an answer to this issue.

Since Security Assertion Markup Language (SAML) [2] depends on the browser agent to broker the authentication transaction, it is primarily utilized as a Web-based authentication technique. Figure 7 depicts the SAML authentication flow at a high level.

Based on LDAP attributes and SOLID decentralized data stores for User Data, SOLID PODs are used by the Cloud Data Storage Tier to hold user-related data and attributes connected to the LDAP protocol, which are essentially key-value pairs. SOLID data stores are decentralized, compatible data repositories that follow guidelines that support user ownership and control over their data. User profiles, access permissions, login passwords, and other pertinent information needed for identity management are safely stored on these PODs. The settings, directory structures, and schema data required for LDAP protocol-based interactions and activities inside the identity provider system are included in the LDAP-related data. Thanks to SOLID PODs, we can really achieve a privacy-preserving identity management service across multi-Clouds, since PODs will migrate across Clouds, ready to be accessed by the same (owner) user, without the need for re-identification in different Clouds. The latter guarantees the identity provider system accessibility, security, and usefulness. The system can be efficiently managed, scaled, and maintained while satisfying the needs of both system administrators and end users.

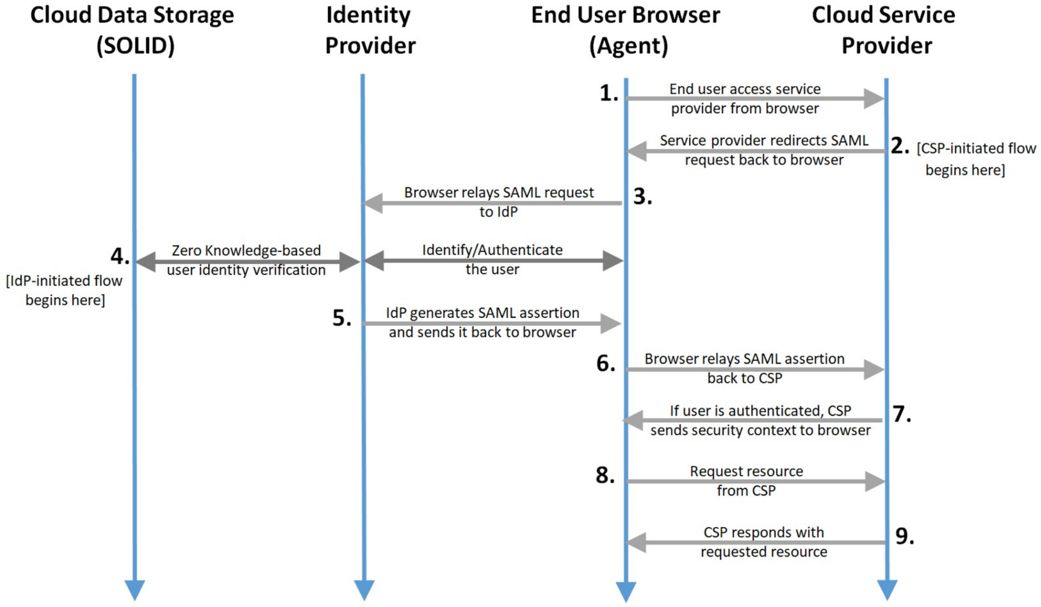
Cloud Data Storage (SOLID)
Identity Provider
End User Browser (Agent)
Cloud Service Provider
1. End user access service provider from browser
2. Service provider redirects SAML request back to browser
[CSP-initiated flow begins here]
3. Browser relays SAML request to IdP
4. Zero Knowledge-based user identity verification
[IdP-initiated flow begins here]
Identify/Authenticate the user
5. IdP generates SAML assertion and sends it back to browser
6. Browser relays SAML assertion back to CSP
7. If user is authenticated, CSP sends security context to browser
8. Request resource from CSP
9. CSP responds with requested resource

Figure 7: Sequence Diagram of Data Exchange and Access.

We present a few often used SAML concepts, as follows. (i) The organization offering the service is known as a Cloud service provider, and they usually take the form of a Cloud-based application. (ii) The organization that supplies the identities, together with the capability of user authentication, is known as an identity provider. The user profile, which includes other details about the user including their job code, address, phone number, and first and last names, is usually also included in the Identity Provider. Certain service providers could need a very basic profile (username, email), while others might need a more comprehensive collection of user data (job code, department, address, location, manager, and so on), depending on the application. (iii) The Cloud service provider creates a SAML Request, sometimes called an authentication request, in order to “request” an authentication. (iv) The identity provider generates a SAML Response. It includes the verified user genuine claim. Depending on what the Service Provider can offer, a SAML Response may also include other data, such as user profile and group/role information. (v) When a Cloud service provider initiates the SAML sign-in flow, it is referred to as a CSP-initiated sign-in. This usually happens when the end user tries to sign in or access a resource directly from the Service Provider end, as when the browser tries to access a resource that is protected from access from the Service Provider end. (vi) An identity provider-initiated sign-in (IdP-initiated sign in) denotes the SAML sign-in process that the Identity Provider started. In this flow, the Identity Provider starts a SAML Response that is routed to the Service Provider in order to confirm the user identity, as opposed to the SAML flow being started by a redirection from the Service Provider.

3. Conclusions and Future Work

In conclusion, digital identity management services play a crucial role in Cloud Computing infrastructures. Their role is to authenticate users, facilitate flexible access control to services based on user identity features, and safeguard data privacy. The proposed methodology aims to enhance interoperability across various domains while simplifying identity verification management in a privacy-preserving manner. This is achieved through the utilization of high-level identity verification policies, such as identity attributes, zero-knowledge proof protocols, and semantic matching techniques. Additionally, decentralized secure data stores are employed to ensure data security. The strength of our proposal lies in the well-established SOLID PODs concept.

Future work primarily focuses on enhancing our framework by incorporating cutting-edge features in big data processing and management in Clouds (e.g., [22,23,25,26,27,28]), as this relevantly impacts on the performance of the overall framework.

Acknowledgements

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union -NextGenerationEU.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, L. B. Othmane, and L. Lilien. "An Entity-Centric Approach for Privacy and Identity Management in Cloud Computing". In: *SRDS 2023, 29th IEEE Symposium on Reliable Distributed Systems*, pp. 177–183, 2010.

[2] A. Armando, R. Carbone, L. Compagna, J. Cuellar, L. Tobarra. "Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-Based Single Sign-On for Google Apps". In: *FMSE 2008, 6th ACM Workshop on Formal Methods in Security Engineering*, pp. 1–10, 2008.

[3] E. Bertino, F. Paci, R. Ferrini, N. Shang. "Privacy-Preserving Digital Identity Management for Cloud Computing". *IEEE Data Engineering Bulletin 32(1)*, pp. 21–27, 2009.

[4] T. Chaudhary, S. Kalra. "Interoperable Identity Management Protocol for Multi-Cloud Platform". *International Journal of Big Data Intelligence 6(2)*, pp. 69–85, 2019.

[5] J. Cui, X. Zhang, H. Zhong, J. Zhang, L. Liu. "Extensible Conditional Privacy Protection Authentication Scheme for Secure Vehicular Networks in a Multi-Cloud Environment". *IEEE Transactions on Information Forensics and Security 15*, pp. 1654–1667, 2020.

[6] M. Deng. "Privacy Preserving Content Protection (Privacy Behoud Content Protection)". *Faculty of Engineering Katholieke Universiteit Leuven, Leuven, Belgium*, 2010.

[7] C. Emig, F. Brandt, S. Abeck, J. Biermann, H. Klarl. "An Access Control Metamodel for Web Service-Oriented Architecture". In: *ICSEA 2007, 2nd IEEE International Conference on Software Engineering Advances*, art. 57, 2007.

[8] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli. "Proposed NIST Standard for Role-Based Access Control". *ACM Transactions on Information and System Security 4(3)*, pp. 224 –274, 2001.

[9] O. Goldreich, H. Krawczyk. "On the Composition of Zero-Knowledge Proof Systems". *SIAM Journal on Computing 25(1)*, pp. 169–192, 1996.

[10] T. Howes, M. Smith, G.S. Good. "Understanding and Deploying LDAP Directory Services". *Addison-Wesley Professional*, 2003.

[11] A. Jøsang, J. Fabre, B. Hay, J. Dalziel, S. Pope. "Trust Requirements in Identity Management". In: *AusGrid 2005, 2005 Australasian Workshop on Grid Computing and E-Research*, pp. 99–108, 2005.

[12] A. Jøsang, S. Pope. "User Centric Identity Management". In: *APSIRC 2005, AusCERT Asia Pacific Information Technology Security Conference*, 2005.

[13] P. Kumar, V.K. Sehgal, D.S. Chauhan, P. Gupta, M. Diwakar. "Effective Ways of Secure, Private and Trusted Cloud Computing". *CoRR abs/1111.3165,* 2011.

[14] A. Cuzzocrea, I. Belmerabet. “Towards Privacy-Preserving Multi-Cloud Identity Management using SOLID”. In: *SECRYPT 2024, 21st International Conference on Security and Cryptography,* pp. 649–654, 2024.

[15] M. Neuenschwander. “Enterprise Identity Management Market 2006–2007”. *Burton Group Identity and Privacy Strategies*, 2006.

[16] P.S. Pawar, A. Sajjad, T. Dimitrakos, D.W. Chadwick. “Security-as-a-service in Multi-Cloud and Federated Cloud Environments”. In: *IFIPTM 2015, Trust Management IX, 9th IFIP WG 11.11 International Conference*, pp. 251–261, 2015.

[17] S.K.S. Raja, A. Sathya, S. Karthikeyan, T. Janane. “Multi Cloud-Based Secure Privacy Preservation of Hospital Data in Cloud Computing”. *International Journal of Cloud Computing 10(1-2)*, pp. 101–111, 2021.

[18] SOLID, Available at: https://solidproject.org/, 2024

[19] G. Spyra, W.J. Buchanan, E. Ekonomou. “Sticky Policy Enabled Authenticated OOXML”. In: *SAI 2016, SAI Computing Conference*, pp.1118–1122, 2016.

[20] Z. Tari. “Security and Privacy in Cloud Computing”. *IEEE Cloud Computing 1(1)*, pp. 54–57, 2014.

[21] R. Weingärtner, C.M. Westphall. “Enhancing Privacy on Identity Providers”. In: *SECURWARE 2014, 8th International Conference on Emerging Security Information Systems and Technologies*, pp. 1–7, 2014.

[22] R.C. Camara, A. Cuzzocrea, G.M. Grasso, C.K. Leung, S.B. Powell, J. Souza, B. Tang. “Fuzzy Logic-Based Data Analytics on Predicting the Effect of Hurricanes on the Stock Market”. In *FUZZ-IEEE 2018, 2018 IEEE International Conference on Fuzzy Systems,* pp. 1–8, 2018.

[23] P. Howlader, K.K. Pal, A. Cuzzocrea, S.M. Kumar. “Predicting Facebook-Users’ Personality Based on Status and Linguistic Features via Flexible Regression Analysis Techniques”. In: *SAC 2018, 33rd Annual ACM Symposium on Applied Computing,* pp. 339–345, 2018.

[24] C.K. Leung, P. Braun, A. Cuzzocrea. “AI-Based Sensor Information Fusion for Supporting Deep Supervised Learning”. *Sensors 19(6),* art. 1345, 2019.

[25] M.J.H. Faruk, H. Shahriar, M. Valero, F.L. Barsha, S. Sobhan, M.A. Khan, M.E. Whitman, A. Cuzzocrea, D. Lo, F. Wu. “Malware Detection and Prevention using Artificial Intelligence Techniques”. In *BigData 2021, 2021 IEEE International Conference on Big Data,* pp. 5369–5377, 2021.

[26] M. Masum, H. Shahriar, H. Haddad, M.J.H. Faruk, M. Valero, M.A. Khan, M.A. Rahman, M.I. Adnan, A. Cuzzocrea, F. Wu. “Bayesian Hyperparameter Optimization for Deep Neural Network-Based Network Intrusion Detection”. In: *BigData 2021, 2021 IEEE International Conference on Big Data,* pp. 5413–5419, 2021.

[27] J.D. Roberts, J.F. DeFranco, D.R. Kuhn. “Data Block Matrix and Hyperledger Implementation: Extending Distributed Ledger Technology for Privacy Requirements”. *Distributed Ledger Technologies: Research and Practice 2(2)*, pp. 1–11, 2023.

[28] A. Kanimozhi, N. Vimala, “Adaptive Weighted Support Vector Machine Classification Method for Privacy Preserving in Cloud over Big Data Using Hadoop Framework”. *Multimedia Tools and Applications 83(2)*, pp. 3879–3893, 2024.