Providing Secured Access Delegation in Identity Management Systems

Abubakar-Sadiq Shehu[1,2][a], António Pinto[3][b] and Manuel E. Correia[2,3][c]
[1]*Department of Information Technology, FCSIT, Bayero University Kano, Kano, Nigeria*
[2]*Department of Computer Science, Faculty of Science, University of Porto, Porto, Portugal*
[3]*CRACS & INESC TEC, Porto, Portugal*

Keywords: Service Providers, Identity Provider, Authentication, Authorisation, Opend ID Connect, Attribute based Access Control, Public Key Cryptography.

Abstract: The evolutionary growth of information technology has enabled us with platforms that eases access to a wide range of electronic services. Typically, access to these services requires users to authenticate their identity, which involves the release, dissemination and processing of personal data by third parties such as service and identity providers. The involvement of these and other entities in managing and processing personal identifiable data has continued to raise concerns on privacy of personal information. Identity management systems (IdMs) emerged as a promising solution to address major access control and privacy issues, however most research works are focused on securing service providers (SPs) and the services provided, with little emphases on users privacy. In order to optimise users privacy and ensure that personal information are used only for intended purposes, there is need for authorisation systems that controls who may access what and under what conditions. However, for adoption data owners perspective must not be neglected. To address these issues, this paper introduces the concept of IdM and access control framework which operates with RESTful based services. The proposal provides a new level of abstraction and logic in access management, while giving data owner a decisive control over access to personal data using smart-phone. The framework utilises Attribute based access control (ABAC) method to authenticate and authorise users, Open ID Connect (OIDC) protocol for data owner authorisation and Public-key cryptography to achieve perfect forward secrecy communication. The solution enables data owner to attain the responsibility of granting or denying access to their data, from a secured communication with an identity provider using a digitally signed token.

1 INTRODUCTION

The continuous growth in Internet technologies has increased the adaption to Web enabled services, which eases access and processing of electronic data in a wide range of fields. For example it is used to access social services in healthcare, finance, insurance and educational institutions. However, this advancement has proliferated the risk of users private data exposure to third parties (SPs and IdPs) that manages the data as they (user) barely have control of their data on different services. To ensure users data privacy and regulate third-party access, IdMs with access delegation methods are used.

Access delegation is akin to power of attorney. It is a process of entrusting or transferring acting powers to a legal entity (person, business or application) to act on behalf of another entity in conducting transactions. Electronically, access delegation is achieved in twofold by; authentication and authorisation. Authenticating an entity implies proving that the entity is indeed who they claim to be by presenting what they have or who they are (Jin et al., 2012). While authorisation indicates a decision on what resources an authenticated entity is allowed to perform (Tschofenig, 2015). Some standard protocols used by IdMs in achieving these include, SAML, OAuth, LDAP and Shibboleth.

[a] https://orcid.org/0000-0002-2894-6434
[b] https://orcid.org/0000-0002-5583-5772
[c] https://orcid.org/0000-0002-2348-8075

Shehu, A., Pinto, A. and Correia, M.
Providing Secured Access Delegation in Identity Management Systems.
DOI: 10.5220/0009892206380644
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 638-644
ISBN: 978-989-758-446-6

The ability to delegate access rights is important as it improves on quality and timeliness of service delivery. This process is mostly felt in sectors like health, banking and educational institutions. In Health institutions for example, healthcare practitioners exercises their duty by accessing some electronic data on behalf of the healthcare institution. To do this they would need to access personal health record of a patient and possibly create entries for consultation, which implies a representation of both the Healthcare institution and patient, which gives an uncontrolled access to the institutional and personal health record of the patient (Sá-Correia. et al., 2020). Hence, generates the issue of data privacy

Data privacy is ability to control the release of private data (Gates, 2007). To achieve this by access delegation, a representative has to be properly identified and authorised. In conventional transactions, physical identification with an identity document is sufficient. However, the prevalence of electronic services necessitated the need for an equivalent identification method that supports access delegation with secured gateways. In related works, several IdM concepts using secured protocols where proposed, however many of these works are unable to address the needed flexibility for modular approach to access delegation and allowing data owners determine their own privacy.

To address these issues, we introduce SChEMER (u**S**er **C**entred **M**andate **R**epresentation), for preserving privacy of users personal information on IdMs. The solution is a decentralised and user centred access delegation framework that adopts some technologies; OIDC, ABAC and Public key cryptograpy. It extends a standard authentication and authorisation request of a user and enables a data owner to asynchronously determine appropriate access, authorise and delegate responsibility or share private resources with the user. It provides an additional layer of security, where data owner assumes full control over the disclosure of their identity data through an assertion issued from their mobile phones to an authorisation server which in turn issues an access token.

2 THREAT MODEL

The use of eServices has exposed users sensitive data to third parties whom are not only able to access private data, but also keep a registry of users access pattern. The provision of these services to users irrespective of their location invariably involves data-sharing among domains. A breach of trust can be encountered in a federated protocol where a user gets authenticated to a compromised IdP using an identity credential which is centrally controlled and agreed on by both SP and IdP. Once done the client application copies these credentials and are used to access back end services. In many cases users tend to repeat access credentials on services they access. However, we assume here that different credentials are used on different services, but the client application gets around this by demanding for leading questions that clearly reveals user’s identity, sometimes with promised access to other or extended services.

Some example projects for the interoperability of users data in Europe include: epSOS (cross- border exchange of health data), e-CODEX (cross- border legal services), STORK and eIDAS (regulation on electronic identification and trust services in e-transactions). While these projects are mostly meeting their goals in easing access to interoparable eServices, they have continued to raise issue relating to privacy of users which include but not limited to the following:

1. Non-optional trust on IdPs; Users are coerced to trust IdPs and SPs who request and stores private user information that are more than necessary to access a service. If an IdP suffers a security breach it loses private users data, therefore users rather than the Idp bears the economic consequences of identity theft and invasion of privacy (Sabouri et al., 2012).
2. Data integrity and confidentiality; are SPs able to guarantee that private data is free from unauthorised access and manipulations?
3. Trusted parties colluding to mount an attack on private data. This can be experienced in proxy re-encryption method used in (Dash et al., 2017), where trusted proxy colludes with recipient to generate the originally encrypted message.

3 RELATED WORKS

Possible breech of users private data lies between the receipt, processing and dissemination of the data. With the current role attained by SPs and IdPs, users privacy on their data is non guaranteed. Some works aimed at achieving users privacy on eServices have been produced.

The work in (Leitold et al., 2014), (Zheng et al., 2015) and (Falcão-Reis and Correia, 2010)
takes data regulatory provisions aimed at protecting users private data (Directive, 1995) and (Regulation, 2016) into consideration. These works uses

mechanisms that demand data owner's consent to access personal data, but such consent does not define the extent of use on requested data. Moreso, since users data is at their disposal, noting stops SPs and IdPs from exploiting the data at will. This further makes a privacy-enhanced IdM indispensable. The work in (Dash et al., 2017), supports user consent on releas- ing identity attributes, but relies on trusting the IdP to re-encrypt the authorised attributes. A collusion within trusted parties can lead to a compromise in public keys used for the encryption. Also, most of the studied works do not ensure minimisation of users data in identity verification, giving services more than the required data to verify a person. Also, once a user has granted right of access on their data they cannot withdraw such right, this further gives free reign on users data to third parties.

To overcome these and other issues, there is a need for security methods that implements policies based on data owners decision flexible enough to allow them control or delegate access to their data seamlessly and at their will. With these data owners would have full knowledge of access on their private data.

4 PROPOSED SOLUTION

We propose a user centred access delegation method using a fine grained access control model to authenticate users. Once authenticated, data owners can issue their consent to allow or deny access to private resources. If allowed, users can carry out only functions delegated by data owner. An example of controlled access delegation is a valet key, where a car owner is able to grants access to doors, trunk, or car safe without the ignition.

The ubiquitous use of smart-phones has created a dynamic computing platform, making it a *de facto* carrier routinely by owners Therefore, integrating owners mobile phone as a platform to authenticate their identity and issue their consent will be seamless and remove the need to trust third parties at different location.

While IdMs simplifies shared authentication within and across domains, it does not include authorisation. Therefore, SChEMER is be based on a combination of OIDC federated identity, ABAC and Public-key cryptography (asymmetric).

4.1 Underlying Technologies

OIDC is an authorisation protocol built upon the OAuth 2.0 framework (Richer et al., 2017), it is used to specify secure access authorisation and delegation methods for users in federated IdMs based on the authentication performed by an authorisation Server (AS). OIDC's AS issues cryptographycally secured and signed credential access and ID token, which are used to verify a request and contain several claims and information about users request (e.g attributes, unique identifiers and other meta data).

To acquire an access token, **(1)** a user requests to access some protected resources from an SP. Since the user is unknown to the SP, it responds with a redirect URI containing information needed to verify, authenticate and request for user authorisation at a delegated IdP. **(2)** User then authenticates at the IdP, using certified credentials. **(3)** Once authentication and authorisation is completed the IdP responds with authorisation code, implicit code or user credentials. To enhance user privacy and protect against hijacking of authorisation response, protected resources are not delivered directly in plain text rather a code is issued (we adopt the use of authorisation code flow). This flow issues an authorisation code to confirm the users credential and request for consent to share this information with SP before the access token is issued. **(4)** The user then shares the code with the SP. **(5)** SP forwards it to the IdP as a confirmation and re- quest for access and ID token. **(6)** Once received, the SP uses the access token to collect user information from IdP, whereas the ID token is used to confirm the user's identity. **(7)** With the access token, the SP de- livers the IdP's consent to either grant or deny access to the user. To prevent against replay attack and ensure data integrity, OIDC token contains a nonce and expiration timestamp for user's access to protected resources. Together with the token, some hash value of the request information is also generated and disseminated in the response. Once user identity is verified and authorised, the SP responds to the user.

ABAC, is a logical access control mechanism that is based on XACML standard. It employs the use of fine-grained contextual rules to determine the authenticity of a request. Permissions are then granted through the use of policy definitions that are made up of collective attributes of subject and object such as; who, what, why, when, where, how and in what HTTP mode (POST, UPDATE and DELETE e.t.c) are resources accessed (Hu et al., 2014). Access policies define conditions (predicates over attributes) of granting access to requests. ABAC addresses some of the limitations posed by Role based access control (RBAC) framework, like inadequacy in role granularity which can lead to role explosion. ABAC also supports the multi factor

Table 1: Diffie–Hellman key exchange.

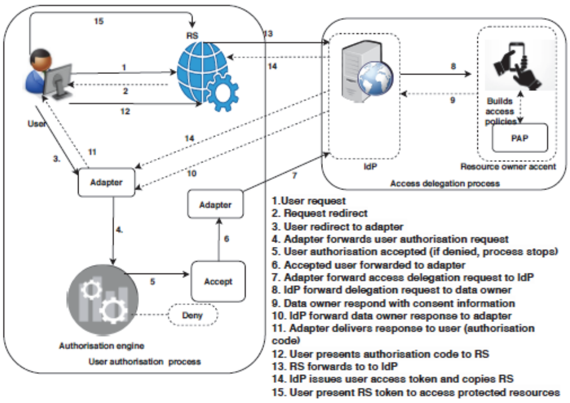
Public Key Cr **eation**		
Communicating entities	**A**	**B**
Step 1: keys selection	*P* and (*g modulo P*)	*P* and (*g modulo P*)
Step 2: integer selection	integer a.	integer b.
Step 3: Computation	g^a *modulo P.*	g^b *modulo P.*
Step 4:Public key	A and B exchange computation in **Step 3**.	
Step 5: Secret keys	$(g^b)^a$ *modulo P.*	$(g^a)^b$ *modulo P.*
Step 6: Key exchange	A and B exchange keys in **Step 5**.	
Step 7: Secret keys	The shared secret of A and B are equivalent.	

evaluation of users attributes at run-time to determine allowable operations for ex- ample, decision that depends on some or all of users rank, organisation, geo-location, affiliation and access history (Hu et al., 2014) can be made at run-time. In ABAC, user request and access decisions can be de- termined by simply changing attribute values, without the need to change the entities relationship defining underlying policy. Additionally, ABAC presents great advantage in access control as it encompasses the concepts of Mandatory Access Control (MAC), Discretionary Access Control (DAC) and RBAC (Ausanka-Crues, 2001; NIST, 1995). ABAC meets the required flexibility of an access control model, as any attributes that identifies an entity can be used to create rules (Hu et al., 2014). It also fit to be applied externally on APIs, databases and other secured resources and supports high performance even in complex environments. XACML defines the standard architecture for the implementation of ABAC. A XACML reference architecture is made up of: **(1)** Policy enforcement point (PEP); located at a web server, it intercepts a request, translates to XACML and sends to Policy decision point (PDP). PEP enforces authorisation decisions at the resource server (RS). **(2)** PDP; Receives XACML request from PEP, evaluates and propagates request with respect to attribute and policies. **(3)** Policy information point (PIP); Serves as the repository holding attribute information about subject and objects. It also serves as the sole policy repository with default policies and attributes. **(4)** Policy administration point (PAP); stores the abstract authorisation policy in databases example, LDAP and SQL. It anchors scopes and policies to PDP.

Public-key cryptography (Hankerson et al., 2006) is a key exchange method where a common key pair is used as a secret key between two communicating parties in achieving perfect forward secrecy communication. To arrive at a common key pair the communicating parties first select two keys, a public key that is known across the communicating channel and a private key known only to the owner. For example entities A and B agrees on a large prime number *p* and a nonzero integer *g* to compute a public key *(g modulo p)*. Both parties then choose secret integers a and b, and exchange a message with value g^a *mod- ulo p* and g^b *modulo p* respectively. With this they are able to compute their mutual secret keys without knowing their private keys; A computes $(g^b)^a$ *modulo p* while B computes $(g^a)^b$ *modulo p*. The values computed by A and B respectively are actually the same, since $(g^b)^a$ *modulo p* = $(g^a)^b$ *modulo p*. The common value generate is their exchanged key. With this key they will be able to encode their message pri- vately since it is know to them only. It addresses the key management and distribution issues in asymmet- ric key cryptography by ensuring data confidentiality between communicating parties (Kuegler and Sheffer, 2012). An example of this is the Diffie Hellman key exchange protocol as illustrated in *Table* 1.

4.2 Description of SChEMER

SChEMER is a user centred access delegation framework aimed at ensuring privacy of users data. This model is supported by a Mobile and Web application, it is made up of the following entities; **(1)** User; a person requesting access to protected resources at RS, **(2)** User authorisation engine; an authorisation endpoint for user based on ABAC. **(3)** RS; Protected resources are stored on this server; **(4)** IdP; OIDC provider, which contains an AS that encapsulates the endpoints for data owner's authentication and authorisation. **(5)** Data owner; a person that owns private resources. In contrast to user managed access (UMA) (Richer et al., 2017), where data owner's policy on private resources is pre-defined, our solution enables data owner to pro- vide access policies, consent and access delegation method on protected resources when an access re- quest is made, which can be delivered asynchronously on their mobile phone. To do this, data owner receives a notification informing them on the need to respond to an access request. Before giving consent, data owner is able to verify users attributes and determines access scope. SChEMER uses OIDC's authorisation code flow to asynchronously issue access token after a data owner's consent is received from their mobile phone. The use of authorisation code via URI query prevents the exposure of browser parameters (example cache, log files) and replay attacks. It also prevents phishing attacks, since the actual access token is not revealed. The components in SChEMER are divided into two authorisation sections as shown in figure 1, which are connected via the RS and IdP through an adapter. The first section performs user authorisation based on internal policies to confirm



1.User request
2. Request redirect
3. User redirect to adapter
4. Adapter forwards user authorisation request
5. User authorisation accepted (if denied, process stops)
6. Accepted user forwarded to adapter
7. Adapter forward access delegation request to IdP
8. IdP forward delegation request to data owner
9. Data owner respond with consent information
10. IdP forward data owner response to adapter
11. Adapter delivers response to user (authorisation code)
12. User presents authorisation code to RS
13. RS forwards to to IdP
14. IdP issues user access token and copies RS
15. User present RS token to access protected resources

Figure 1: Example instantiation of the architecture.

the authenticity of user and the resources requested for. This re- quest is captured by PEP server within the authorisation engine, which extracts both user and protected resources attributes for the evaluation and authorisation process within the engine. The second authorisation section is initiated after receiving a request from an adapter which is either completed for first time from authorisation engine or a subsequent request which has already been issued an authorisation code/refresh token. The adapter is used in cross conversion of user authorisation request and access authorisation results into access delegation request and vice-versa, so that they can be forwarded to data owner as an access delegation request and to user as a response. The concept of this work on empowering data owners to decide their privacy and determine access rights at their will tallies with the EU regulations on Data Protection Directive (DPD) and the General Data Protection Regulation (GDPR) These regulations compels third-access request may contain user ID or name, refresh token, and user profession or affiliation. All attributes are evaluated against state policies (for example who may access what resources according to state legislation), organisational and resource policies which are in the policy and attribute repositories. For example to be affiliated with a project, reside in a supported location, possess some qualification or even have some working experience. Therefore, to achieve this authorisation the engine fetches these policies within it from PIP and PAP. Once all attributes for policy evaluation have been gathered and evaluated, the authorisation engine's response is forwarded either to parties to seek explicit owner's consent before using private data. The regulation further mandates all applications, services and entities involved in processing individual data to provide owners with full knowledge of these processes.

4.3 SChEMER Integration with Open ID and ABAC

This section describes the major components of SChEMER framework; which integrates OIDC and ABAC. The framework assumes that data owner is pre-registered at IdP to receive an access delegation request by notification, but unknown to SP. Likewise, the user is known to the SP but not IdP. The interaction process begins with user's request.

User's Request: a user requests for access on protected resources using a client application on behalf of the owner from RS which operates with RESTful service. The RS notices that the user is unknown, so it replies with a redirect to an authorisation engine through an adapter for the user to be authorised. The RS passes along a callback URL (a redirect URL) as a query parameter, which the adapter will use when the authorisation process is completed. The user invokes the authorisation engine via the adapter for authorisation, which provide an authentication and authorisation interface with scopes for user to provide the required credentials. At the authorisation engine the user credential is captured and the request is translated into XACML arbitrary language.

Authorisation Engine Process: At authorisation engine, PEP intercepts user's request, translates and forwards it internally for the user authorisation based on attributes and access policies of the user and the resource been requested. The authorisation engine captures the request that contains user's and protected resources attributes. Depending on the scenario, an adapter that translate and forwards it to IdP or to a terminal end if the authorisation is not granted. The adapter passes along the request information as a query parameter which the IdP will use when user consent is received.

IdP Evaluation: IdP captures the request forwarded by the adapter and sends it to data owner for informed consent. Since SP is only known to the IdP and not data owner, a request from IdP with user's redirect information and authorisation decision guarantees that the user is pre-authorised. Data owner receives this request as a notification on their mobile phone. To issue informed consent, data owner needs to sign into the IdP server and be authenticated, we assume that a client application that enables user authenticate to IdP is pre-installed on the data owners mobile phone. Once authenticated they will be able to respond asynchronously to requests by either granting or denying it. Before issuing this consent, both data owner and IdP needs some assurance that the right owner has received the request, to do this a secured secret key generation process is initiated using the Diffie Hellman public key exchange discussed in *Section* 4.1, with which they both generate common secured key to confirm their identities. Once both parties are able to confirm their identities and establish a common secured key, the data owner drafts a delegation policy with the client application using the PAP, and forwards a response to the IdP. The IdP computes an authorisation code with query parameters in the URL and sends it to the user via the adapter. To access the resources the user presents the authorisation code to the SP in exchange for a digitally signed OAuth JSON Web access Token (JWT) and ID token at the IdP, that is only understood by the RS but opaque to client application and user. The RS submits the authorisation code directly to the IdP for confirmation of code, users authorisation process and scope of access. IdP then responds with the access, refresh and ID token. The access token is used to invoke RS forthe protected resources. While the ID token contain set of claims about the authentication session such as user, IdP and client application ID, and validity of the token. In other to protect an attacker from overcoming IdP's security, data owner issues an access token that contains policy and scope, which determines access lifetime, purpose, method, location, usage and ability to revoke the token (both refresh and access). With access token a user is able to further invoke the RS for the purported service at the same instance, while a refresh token is used to access the same resource within the lifetime of the access token.

5 CONCLUSIONS AND FUTURE WORK

This paper introduces a user centred access delegation framework. It foresees a method that secures users privacy and ensure data confidentiality by authenticating a requestor, and granting only an authorized requestor access to data via a revocable token. This manifestation has detached the need to trust an external IdP residing at the SP or controlled by third parties and vice versa.

Being part of a work in progress, we strongly rely on already implemented IdMs, client applications and Government owned registers for integrating the method. For future development, we plan to implement our framework within the health care, education and other social services to support seamless interoperability of citizens data. This we believe will further support EU digital single market.

ACKNOWLEDGEMENTS

This work is partially financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020."

REFERENCES

Ausanka-Crues, R. (2001). Methods for access control: advances and limitations. *Harvey Mudd College*, 301:20. Dash, P., Rabensteiner, C., Hörandner, F., and Roth, S. (2017). Towards privacy-preserving and user-centric identity management as a service. *Open Identity Sum- mit 2017*.

Directive, E. (1995). 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the EC*, 23(6).

Falcão-Reis, F. and Correia, M. E. (2010). Patient empowerment by the means of citizen-managed electronic health records. *Medical and Care Compunetics*, 6:214–228.

Gates, C. (2007). Access control requirements for web 2.0 security and privacy. *IEEE Web*, 2(0).

Hankerson, D., Menezes, A. J., and Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer Science & Business Media.

Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., and Scarfone, K. (2014). Guide to attribute based access control (abac) definition and considerations. *NIST Special Publication*, 800:162.

Jin, X., Krishnan, R., and Sandhu, R. (2012). A unified attribute-based access control model covering dac, mac and rbac. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 41–55. Springer.

Kuegler, D. and Sheffer, Y. (2012). Password authenticated connection establishment with the internet key exchange protocol version 2 (ikev2). *IETF RFC 6631*.

Leitold, H., Lioy, A., and Ribeiro, C. (2014). Stork 2.0: Breaking new grounds on eid and mandates. In *Proceedings of ID World International Congress*, pages 1–8. .

NIST, N. (1995). An introduction to computer security: The nist handbook. *NIST Special Publication*, pages 800–12.

Regulation, E. (2016). 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *European Union*, pages 1–88.

Richer, J., Sanso, A., and Glazer, I. (2017). *OAuth 2 in Action*. Manning Publications.

Sabouri, A., Krontiris, I., and Rannenberg, K. (2012). Attribute-based credentials for trust (abc4trust). In *International Conference on Trust, Privacy and Security in Digital Business*, pages 218–219. Springer.

Sá-Correia., L., Correia., M. E., and Cruz-Correia., R. (2020). Illegitimate his access by healthcare professionals detection system applying an audit trail-based model. In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5 HEALTHINF: HEALTHINF,*, pages 539–546. INSTICC, SciTePress.

Tschofenig, H. (2015). Oauth working group j. bradley internet-draft ping identity intended status: Standards track a. sanso, ed. expires: January 22, 2016 adobe systems.

Zheng, H., Yuan, Q., and Chen, J. (2015). A framework for protecting personal information and privacy. *Security and Communication Networks*, 8(16):2867–2874.