IJRIT
Always Searching for Research

International Journal of Research in Information Technology (IJRIT)

www.ijrit.com ISSN 2001-5569

A Simple Distributed Identification Protocol for Triplestores

E. Arun Kumar Goud [1] , M. Janga Reddy [2]

[1] M.Tech student, Department of CSE, CMR institute of Technology, Dist. R.R, Hyderabad, AP, India
Email-id: arunkumar0078@gmail.com,

[2] Professor and principal, Department of CSE, CMR institute of Technology, Dist. R.R, Hyderabad, AP and India
Email-id: principalcmrit@gmail.com

Abstract

OAuth is an open standard for authorization. OAuth provides a method for clients to access server resources on behalf of a resource owner. It also provides a process for end-users to authorize third-party access such as games, and productivity applications access to user online private data to their server resources without sharing their credentials, using user-agent redirections. In this paper defines a simple identification protocol dedicated to triplestores which is universal and appropriate for the distributed environment. We propose a mechanism based on the HTTP standard, extended with OAuth Protocol and Semantic Web ontology. One can optionally adopt Transport Layer Security protocol. Here we present a scalable method that permits user authentication and authorization to triplestores with data integrity and confidentiality.

Keywords-- Authentication, authorization, access control list, triple store.

I. Introduction

OAuth is associate degree open customary for authorization. OAuth provides a way for purchasers to access server resources on behalf of a resource owner. It additionally provides a method for end-users to authorize third-party access to their server resources while not sharing their credentials sort of a username and secret try, victimization user-agent redirections. OAuth may be a service that's complementary to, and so distinct from, Open ID.

In this paper we describe a triplestore is a purpose-built database for the storage and retrieval of Resource Description Framework (RDF) data.Much like in the case of other databases; one can find and modify data in triplestores via a query language, such as SPARQL Protocol and RDF Query Language. An RDF triple consists of a subject, a predicate, and an object. The subject denotes the resource, the predicate means traits or aspects of the resource, and expresses a relationship between the subject and the object. A collection of RDF statements intrinsically represent a labeled, directed multigraph. The nodes are the subjects and objects of their triples. In the context of triplestores there is a need to provide security means similar to those specific to classical databases. In this paper a new distributed identification mechanism based on OAuth protocol and access control ontology for triplestores is presented. OAuth allows users to share private resources stored on one site with another site without having to hand out their credentials, typically login and password.

II. Literature Survey

Lujun Fang et al. [1] projected a example for the planning of a social networking privacy wizard. The intuition for the planning comes from the observation that real users conceive their privacy preferences that friends ought to be able to see that data supported associate degree implicit set of rules. Thus, with a restricted quantity of user input, it's sometimes doable to make a machine learning model that briefly describes a specific user's preferences, and so use this model to piece the user's privacy settings mechanically. This methodology follows 2 necessary things. First, real users tend to conceive their privacy preferences in terms of communities, which may simply be extracted from a social network graph victimisation existing techniques. Second, our active learning wizard, victimization communities as options, is ready to suggest high-accuracy privacy
Settings victimisation less user input than existing policy-specification tools. This model, then, is
employed to mechanically piece the user’s elaborate privacy settings.

Alessandro Acquisti et al. [2] discussed a representative sample of the members of the Facebook a social network for colleges and high schools at a US academic institution, and compared the survey data to information retrieved from the network itself. And looked for underlying demographic or behavioral differences between the communities of the network’s members, non-members and analyzed the impact of privacy concerns on members’ behavior; compare members’ stated attitudes with actual behavior. In this study extended that an individual’s privacy concerns are only a weak predictor of his membership to the network. Also privacy concerned individuals join the network and reveal great amounts of personal information. Some manage their privacy concerns by trusting their ability to control the information they provide and the external access to it.

Dr. Carrie E. Gates [3] made public personal data presently tends to be loosely outlined by legislation, instead of by what people envisage to be personal. Generic data like a human home address and signal area unit ordinarily thought of in person diagnosable data (PII) and area unit to be protected once collected and keep by a company in addition, the employment and unleash of specific information, like money or medical data, is controlled legislatively. However, there additionally exists data that a personal might envisage to be personal, and wish to unleash solely to specific individuals or individuals meeting a specific criterion. Therefore an individual would possibly wish management to regulate parts of their digital life within the same manner that they
Control what data is discharged in their analog life. Within the analog world, an individual will value more highly to tell somebody or some cluster some piece of knowledge regarding them. However, it's usually the case that within the on-line world these controls don't exist, resulting in actual public revealing.

Gediminas Adomavicius et al. [4] provide an overview of the class of multi-criteria recommender systems. First, it defines the recommendation problem as a multi criteria decision making (MCDM) problem, and reviews MCDM methods and techniques that can support the implementation of multi-criteria recommenders, discuses multi-criteria rating recommenders’ techniques that provide recommendations by modeling a user’s utility for an item as a vector of ratings along several criteria. A review of current algorithms that use multi-criteria ratings for calculating predictions and generating recommendations is provided. In most recommender systems, the utility function usually considers a single criterion value, e.g., an overall evaluation or rating of an item by a user. In general, this assumption has been considered as limited, because the suitability of the recommended item for a particular user may depend on more than one utility-related aspect that the user takes into consideration when making the choice. Particularly in systems where recommendations are based on the opinion of others, the incorporation of multiple criteria that can affect the users’ opinions may lead to more accurate recommendations. Thus, the additional information provided by *multi-criteria ratings* could help to improve the quality of recommendations because it would be able to represent more complex preferences of each user.

Hannes Tschofenig et al. [5] derived the Open Authorization (OAuth) protocol permits a user to grant a third party information processing system or application access to the user’s protected resources, while not essentially revealing their long-run credentials, or perhaps their identity. For example, a photosharing website that supports OAuth might enable its users to use a third-party printing information processing system to print their personal footage, while not

permitting the printing website to achieve full management of the user's account. OAuth may be a fairly versatile protocol which will be deployed by third party websites yet as by downloadable applications on finish devices. The presently current add the IETF OAuth working party standardize the core parts, whereas alternative components area unit left for additional work like token encryption and token content or area unit outside the scope of IETF standardization. the subsequent enhancements to the browser would be useful for safer OAuth deployments, yet as for safer preparation of alternative identity and authorization frameworks a number of the options are common and re-usable.(i) Authentication Mechanisms (ii) Authorization
Interface (iii) Standardized JavaScript Crypto Library Support (iv) Moving Crypto Into the Browser.

Andrew Besmer et al. [6] explored the appliance of social navigation to access management policy configuration using associate degree empirical between subjects study. Social navigation might aid users in creating higher choices by informing them of the previous choices created by themselves or others. Social navigation is outlined because the use of social data to assist a user's call. Social navigation is often employed in everyday interactions. In general, an individual would possibly plan to visit a store supported the amount of cars set outside. One will use cues like this to create associate degree interpretation of the attractiveness of the shop. a lot of cars might indicate higher costs or a wider choice, whereas fewer cars might indicate higher costs and a lot of exclusivity. Social navigation may be wont to impact user behavior and their ensuing privacy policies, though that impact could also be tiny. During this and similar domains, social navigation isn't helpful for motivating a lot of users to contemplate their privacy and modify their policies. The cue might solely have a sway on people who area unit already creating policy choices.

III. Distributed Identification Mechanism for Triplestores

This part we discuss the idea of using the identification mechanism as provided as oAth. And we define a new ontology devoted to the specification of access control by means of authorization. In this case where confidentiality is needed optional, encryption based on Transport Layer Security (TLS) can be used.

Authentication and Data Integrity over OAuth

In this Section we suggest using OAuth to access a triplestore. It is token-based authentication. That means that a logged-in user has a unique token used to access data from the triplestore. Users access to triplestore data with sharing tokens and without disclosing any identity data. This approach presents a triplestore as a server. It could be, optionally, an authorization endpoint and/or an access control lists repository. A client is an application that uses OAuth to access the triplestore on behalf of the user.

An authorization algorithm using OAuth over Hypertext Transfer Protocol. It consists of the following seven steps:
1. Obtain request token from the triplestore2 to the client.
2. Redirect client to the authorization endpoint.
3. The server requests user to sign in by using login and password. It is important that in this step login and password should be encrypted.
4. If login and password are correct, the server associates the user with the role and asks for approval granting to triplestore.
5. Redirect from the authorization endpoint to the client.
6. Exchange request token for access token.
7. The client is ready to request the private data to triplestore.

OAuth take care of the data integrity by signing HTTP requests. The OAuth protocol is secure, because it has tokens (and the fields: timestamp and nonce) that do not pass login and password, for verifying unique requests. Each token grants access for a specific triplestore resource and for a defined duration.
The main disadvantage of OAuth is that login, password and other settings, such as email, cannot be changed via this protocol. OAuth over HTTP does not provide confidentiality. The solution to this problem is to use Hypertext Transfer Protocol Secure (HTTPS). This proposal does not force the use of HTTPS and allows using HTTP, when confidentiality is not needed. The main disadvantage is that HTTPS requires both parties to the communication to do extra work in exchanging handshakes and encrypting and decrypting the messages, making this form of communication slower than it would be without it.

Authorization: User Access Ontology

In this Section the proposed User Access Ontology is presented (in the sequel denoted by UAO). UAO is an ontology describing roles, their permissions, and allowed or permitted actions on triplestores. It allows the description of access control lists for users, without assigning them to a single triplestore and/or other databases. UAO is a descriptive vocabulary expressed in Resource Description Framework (RDF), RDF Schema and Web Ontology Language (OWL). The presented ontology is written in the RDF/XML and Terse RDF Triple Language syntaxes.

We define an authorization $a \in AuthZ$ as a tuple of the form $< role, action >$ where $role \in R$ and $action \in ACT$. The user description (User class) consists of first name (firstName property), last name (lastName property) and user name (userName property). The most important is user name, because it identifies and associates the authenticated user with access control lists. The user characteristics can be extended to other ontologies, such as FOAF [28]. Users are assigned (hasRole property) to roles (Role class). Users should have a minimum of one role. Roles may have names (roleName property). There is also default policy for the role (DefaultPolicy class), which could deny (Deny class) or permit (Permit class) access to data. It should have exactly one default policy. Roles are assigned (hasPermission property) to their permissions (Permission class). Let P be the permissions, USR be the user and DP be the default policy, then role $R \in usr, p, dp$ with $usr \in USR$, $p \in P$ and $dp \in DP$. Permissions may have numeric priorities (priority property), which prevents conflicts. It should also have filters (filter property), which are sets of triple pattern TP or URI references U (see Section 1). Let V be the set of all variables, then $TP \subset (S \cup V) \times (U \cup V) \times (O \cup V)$ and V is infinite and disjoint from U, B, L and D (see Section 1). Permissions may have named graph declaration (graph property). When this value is not set, the permisions refer to the default graph.

The permissions are assigned (has Action property) to actions (Action class). The actions ACT specify roles which are granted to access the triplestore, as well as what operations are allowed or forbidden on the triplestore. We propose nineteen types of actions, which are based on the SPARQL clauses [1,2] and that can be combined with each other. These types of permission are divided into three groups: graph management (Graph Manage class), graph modification (Graph Modify class) and query forms (Query From class). The graph management permissions allow the execution of the clauses: CREATE and DROP. The graph management permissions allow the execution of the clauses: INSERT DATA, INSERT, LOAD, DELETE DATA, DELETE, DELETE WHERE, CLEAR and DELETE/INSERT. The query form permissions allow the execution of the read-only query forms: SELECT, CONSTRUCT, ASK and DESCRIBE. All SPARQL clauses are reflected in classes.

Example of access control list

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix uao: <http://example.org/uao#> .

_:u01 a uao:User ;
        uao:firstName "John" ;
        uao:lastName "Smith" ;
        uao:userName <http://example.org/card#me> ;
        uao:hasRole _:r01 .

_:r01 a uao:Role ;
        uao:roleName "teachers" ;
        uao:hasDefaultPolicy uao:Permit ;
        uao:hasPermission _:p01 .

_:p01 a uao:Permission ;
        uao:priority "10"^^xsd:int ;
        uao:graph "$g" ;
        uao:filter "($s $p $o)" ;
        uao:hasAction uao:Select .
```

Implementation

We used PHP5 as the development platform. The system consists of the following five parts: query engine, access control lists repository (applying User Access Ontology), protected resources (a part of triplestore), authorization endpoint and HTTP client. Our implementation of the proposed prototype is made of three internal modules, namely query engine, access control lists repository and protected resources. Fig. 1 presents the system architecture.

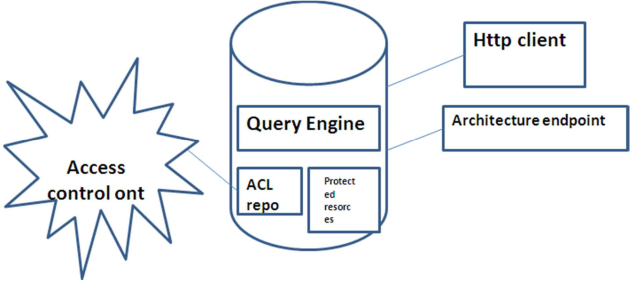

Figure.1. System architecture.

Our modules are additional layers on top of a document-oriented database which substitutes the triplestore. The access control lists and protected resources are stored in triplestore. We use external, third-party OAuth authorization endpoint and a web browser as an HTTP client to test the prototype.

The UML sequence diagram (Fig. 3) presents the workflow of our implementation. The diagram shows basic stages of interaction between actors:

1. The web browser obtains an unauthorized request token.
2. The authorization point and access control list repository authorizes the request token.
3. The web browser exchanges the request token for an access token.
4. The user executes SPARQL queries.

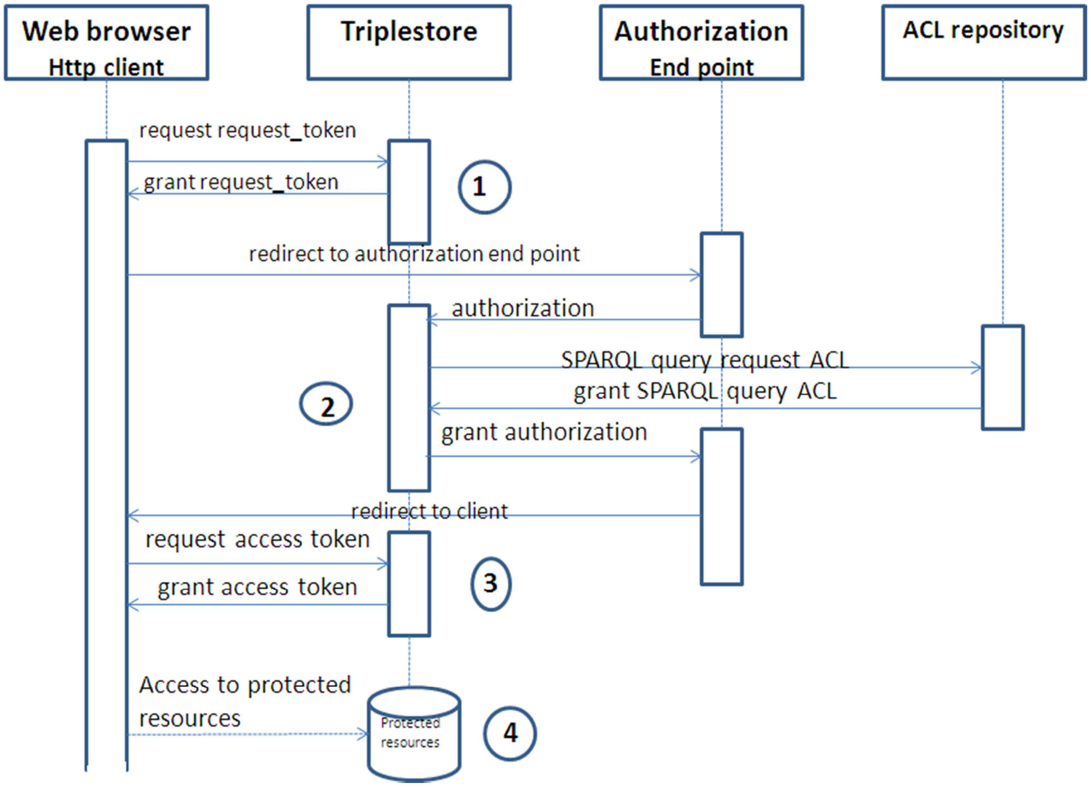

Figure.2. UML sequennce diagram

To enforce an access control, we analyze SPARQL queries and compare to permit or deny actions as exceptions to default policy. Next, the triple patterns from filter and/or graph properties are mapped to the relevant clauses of SPARQL queries. If this wildcard is true for user permissions, the query is executed.

IV Conclusion

We have produced a simple, thought-out RDF standard based and closed triplestores proposal because it is triplestore independent. We have got projected associate identification protocol dedicated to triplestores that is universal and distributed. It uses Protocal, OAuth and ontology. Our proposal will work either with mobile and different devices or a web browser and different software package as a triplestore client and server. Another advantage is that user's don't have to be giving a password to third parties. A crucial advantage of the proposal is that the identification mechanism is distributed and adopts linked data. The implementation shows its great potential.

V References

[1] L. Fang and K. LeFevre, "Privacy Wizards for Social Networking Sites," Proc. Int'l Conf. World Wide Web (WWW), M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, ed., pp. 351-360, 2010.

[2] A. Acquisti and R. Gross, "Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook," Proc. Int'l Workshop Privacy Enhancing Technologies, pp. 36-58, 2006.

[3] D. Carrie and E. Gates, "Access Control Requirements for Web 2.0 Security and Privacy," Proc. Workshop Web 2.0 Security & Privacy (W2SP '07), 2007.

[4] G. Adomavicius and Y. Kwon, "Multi-Criteria Recommender Systems," Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.

[5] Andrew Besmer, Jason Watson and Heather Richter Lipford," The Impact of Social Navigation on Privacy Policy Configuration",Proceedings of the Sixth Symposium on Usable Privacy and Security,Article 7, pages 21--27, 2009.

[6] A. Besmer, J. Watson, and H.R. Lipford, "The Impact of Social Navigation on Privacy Policy Configuration," Proc. Sixth Symp. Usable Privacy and Security (SOUPS '10), July 2010.

[7] Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF, World Wide Web Consortium (2008).

[8] Schenk, S., Gearon P., Passant A.: SPARQL 1.1 Update, World Wide Web Consortium (2010)

[9] Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium (2004).

[10] Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust.
In: 14th International Conference on World Wide Web. ACM, New York (2005).

[11] Hammer-Lahav, E.: The OAuth 1.0 Protocol. Internet Engineering Task Force (2010).

[12] Mykletun, E., Narasimha, M., Tsudik, G.: Authentication and integrity in outsourced databases. ACM Transactions on Storage (2006).

[13] Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L.: Dynamic authenticated index structures for outsourced databases. In: Special Interest Group on Management of Data. ACM, New York (2006).

[14] Bertino, E., Haas, L.M.: Views and security in distributed database management systems. In: Schmidt, J.W., Missikoff, M., Ceri, S. (eds.) EDBT 1988. LNCS, vol. 303, Springer, Heidelberg (1988).

[15] Ahn, G., Sandhu, R.: Role-based authorization constraints specification. ACM Transactions on Information and System Security, TISSEC (2000)