# A Framework for Integrated Threat Detection on Semantic Web Services

Nagendra Kumar Sing, Sandeep Kumar Nayak*
*Email: nksingh444@gmail.com, nayak.kr.sandeep@gmail.com*

## Abstract

*In today's digital age, semantic Web amenities have grown progressively. The increasing use of the Semantic Web has attracted the attention of various users and today they are sharing all their private and confidential information through these amenities. Web amenities such as digital-trade have completely changed the way of doing business and made many revolutionary changes in it. The digital-trade Web amenityhoards assortedhues of data in its database server. It is not feasible to use a single database program by adigital-trade Web amenity to hoard all hues of data because the nature of each data is dissimilar. Therefore it is necessary to develop a mechanism that uses apposite database programs for amassing assorted hues of data. Therefore, it is indispensable that all the information hoarded by the user on the Web server must be fortified against illicit admittance or disclosure of clandestine data to illicit persons. This paper purports the integrated threat prevention framework using dispersed database ploy for digital-trade Web amenity. In this paper, the dispersed database ploy has been recommended to store assorted type of data. The security of a Web amenity has also been augmented by pioneering the concept of IP-MAC trussing for both client and server. When the client and server communicate with each other for the first time, the IP-MAC trussing of the client and server is hoards on both sides. In next conversation, this IP-MAC trussing will be used for authenticating the identity of both client and server. This IP-MAC trussing will also be helpful in preventing the network assails such as Renunciation-of-Amenity (RoA) and Man-in-the-Middle (MITM) on database or Web server.*

***Keywords***: *Access Control, Dispersed database ploy, Digital-trade, Semantic Web, XACML*

## 1. INTRODUCTION

Web amenities are intrinsically embedded in everyone's life. With the continuous growth of Web amenities, it has become possible to access most amenities with the help of internet. Web amenities have been ascertained as key technologies in posing anacquiescent elucidation to interact with a wide variety of application systems. Web amenities can interact with various other Web amenities available on the Internet and barter data and information amid themselves. A Web amenity is a genus of amenity imparted two or more machines to aid the barter of information amid them using the Internet [1]. The Semantic Web is not a newfangled genus of Web although it is an annex of the prevailing Web in which information is delineated profoundly so that Web amenity users can use these amenities facilely. Semantic-annotation has enacted a momentous chore in the augmentation of semantic Web amenities. It plays an important role in finding semantic Web amenities based on their semantics, as well as in commending and crafting semantic Web amenities [2].

The exaggerated form of semantic Web amenities has opened up new avenues of possibilities. A variety of daily-use amenities can be accessed through the Internet via a Semantic Web amenity. The expansion of semantic Web amenities has changed the way we do business today. Semantic Web is also a major reason for the expansion of digital-trade [3]. Digital-trade provides various hues of amenities to its users through the Internet, in which they can acquire information about distinctive products, order products and at the same time users can pay for their products. If seen, digital-trade Web amenity stores and processes many hues of information simultaneously. Digital-trade Web amenity uses Web server to store all these hues of information. Usually the same database program is used to store assorted hues of data and information attainable on digital-trade Web amenities.

For example, digital-trade Web amenities store assorted hues of data such as data related to various products available on digital-trade platform, shopping cart data that accumulates the list of products selected by the customer for purchase, inventory data that holds information regarding the availability of a particular product, user credential data that specifies the user details such as email, phone, login detail, session data, financial data and transactional data etc. If all these hues of data will be managed by a single database program, then digital-trade Web amenity will use a lot of resources to store all these data and this will increase unnecessary processing overhead on the Web amenity. Therefore, the need of the hour is that digital-trade Web amenities should not store assorted hues of data in the same database; rather they should use assorted database programs to store assorted data.

In Semantic Web amenities, communication is mainly done through two servers. The first server is the application server which stores the logic related to the user interface while the second server is the database server which stores the user's data. It is very important to protect both these servers because if an unauthorized user has access to these servers then he can steal and misuse the clandestine information of any user [4]. Unauthorized users make numerous attempts to steal clandestine information and data stored in these Web servers. Primarily Clandestine data remains available on the database server. That is why various network assails such as Renunciation-of-Amenity (RoA) or Man-In-The-Middle (MITM) assails are launched to retrieve clandestine information available on database servers.

Access control is a mode to restrain admittance to the physical or virtual resources of a system. Access control is a process by which legitimate users are given apposite right to admittance system resources, information and meticulous privileges. It is the culpability of the amenity purveyor to preserve the confidentiality of users' data available on the database server and to facilitate users to control access to their personal data [5]. Confidentiality can be demarcated as a suite of classified data that gives rights to the owners of the data to fabricate proclamations allied to admittance, perpetuation, use and transfer of their data. XACML (eXensible Access Control Markup Language) is a policy language for defining admittance restrains policies in a Semantic Web environment [6]. XACML is widely used for specifying access control policies on Web servers. Clandestine information and data available on Semantic Web amenities must be protected through an appropriate access control system.

This paper introduces the integrated threat prevention framework using a dispersed database ploy and IP-MAC trussing for secure semantic Web amenities. This paper proposes a dispersed database ploy for digital-trade Web amenities. Since the digital-trade Web amenity handles assorted hues of data, a sole database suite may not be sufficient to store and process such data. In this paper, a set of database programs is insinuated to handle the assorted type of data. Securing Clandestine data available on database and application servers is also an important issue. To solve this problem, it is insinuated to use a combination of IP-MAC addresses on both the client and server end. The insinuatedploy stores the IP and corresponding MAC address of both client and server during first interaction with both ends. During communication, this IP-MAC trussing will be used in authenticating the identity of both client and server. The insinuated framework aids in ascertaining the data stowage necessities for digital-trade Web amenity. The framework has also made stipulations for sustaining the integrity of data for both client and server.

In this paper efforts are being made to cover the security related issues with Semantic Web amenities.The paper is framed as follows. Section 2 briefly describes the newfangled exploration work in the field of Semantic Web and its security. Section 3 concisely illustrates the insinuated integrated threat prevention framework using dispersed database ploy and IP-MAC trussing. Lastly, the conclusion and future work is given in section 4.

## 2. RELATED WORK

LoukiaKaranikola and IsamboKarali [7] insinuated a framework to deal with the prevalence and incomplete information available on the Semantic Web. The framework defines a fuzzy description of logic by expanding the Dempster – Shaffer theory. The effects of belief in simulated sphere and the Semantic Web as utilized in community network exploration have been reviewed in [8]. The insinuated system uses a virtual world data that logs the punctilious comportments of hefty players and develops semantic Web-based resemblance computing dexterities to scrutinize the hefty data that are being available on social media network. A slant of semantic Web amenityharmonizing instituted on word embedding has been insinuated to extend the process of amenity discovery that elicited with assorted events [9]. The accuracy of QoS (Quality-of-Service) prediction for assuring the quality of a Web amenity has been improved by using the enhanced importance resampling ploy [10]. Pragmatic portrayal manuscripts of Web amenities are archetypally shorter in extent and contain scanter information about the features associated with Web amenities. The classification of any Web amenity cannot be done easily based on the functional detail documents of the Web amenities and hence affects the Web amenity classification. Problems and ambiguities related to Web amenity discovery and their description have been addressed [11]. All detached topographies in the portrayal manuscripts of Web amenities are coalesced to predict the width of the Web amenityassortment by maltreating a comprehensive erudition archetypal. The ambience information of the arguments in the portrayal manuscripts of Web amenitiesisexcavated deployingthe B-LSTM archetypal to deeply predict the Web amenity category. The insinuated methodology uses a rectilinear retrogression tactic to assimilate the width and depth prognostication upshots of Web amenityassemblies as the end upshot of the amenity. The esteem of amalgamated amenities conspicuous the archetype of cross-sphere disseminated amenities from solitary-sphere monumental systems, with the encroachment in the field of cloud computing and the manifestation of amenity marketplaces, which pose generous concealment and security apprehensions. Ratified data exposure and admittance restraint become confront in such systems because substantiation, consent, and data exposer may occur at various points that are not known to its users. A mechanism has been provided to implement security policies in Web amenities for data privacy and security [12]. XACML is an admittance restraint language that outlines innumerable guidelines for employing admittance restraint in Web amenities environment. But as the number and complexity of policies increase, the performance of XACML falls drastically. The authors have stored the XACML policies in a graph database to improve the performance of XACML [13]. The insinuated survey [14] pioneers a set of XACML gages that portray admittance restraint system founded on a supervisor mechanism by aiding coverage yardsticks assortment and on-line pinpointing of tryout activity. The overview of various components of XACML and their performance issues has been provided [15], [16], [17]. Modern applications rely on a multilingual ploy to amassing data, where No SQL database program is used simultaneously with traditional databases. This multilingual ploy is known as polyglot persistence. Polyglot persistence ploy promotes the use of dispersed database programs for amassing diversified data available on a Web amenity. A tool for the instinctive sighting of exterior data contours has been insinuated by scrutinizing the source code of polyglot persistence applications [18]. A conciliation centered constituent has been familiarized to augment and accomplish multifaceted queries on manifold data stockpiles in a cloud milieu [19]. The overview regarding the access control and its implication has been provided [20], [21].

## 3. PROPOSED FRAMEWORK FOR INTEGRATED THREAT DETECTION

XACML (eXtensible Access Control Markup Language) is a platform independent policy language that was developed by OASIS (Organization for the Advancement of Structured Information Standards) [22]. It was developed primarily to define the policies needed to control user admittance to Web amenities. The XCAML language is depleted to restraint user admittance in assorted spheres. It follows the syntax and semantics of XML (eXtensible Markup Language). It mainly consists of four components - Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Administration Point (PAP) and Policy Information Point (PIP).

Whenever a user or subject want to perform some action on protected resources or object, the entreaty of the user or subject moves to the PEP. PEP first ruptures the attributes of the user or subject from the entreaty and transforms the entreaty in XACML format. It forwards the entreaty to the PDP for evaluation. The PDP extracts the applicable policies of the requesting user from PAP and PIP. It then evaluates the entreaty against applicable policies, transfers it decision in the form of accept or deny to PEP. XCAML has emerged as a standard for controlling access in diversified environments such as the Semantic Web, but XCAML also has some trust and privacy loopholes.

The foremost job of XACML is to shield data from illicit users, but it is unable to protect data from compromised users. For example, an unauthorized user is able to crack the user ID and password of a ratified user and uses another device to enter the Web amenity. In this scenario, XACML will only verify the user's credentials, but not the device from which the unauthorized user has inputted the credentials. Thus XACML can allow an unauthorized person to use the Web amenity. This should be clogged by XACML. Therefore, it is required to enforce the provision that the user's other information such as the user id and password as well as the IP and MAC address of the device used by the user should be hoarded when the user registers on the Web server. Whenever a user sends a request to login to a Web amenity, the IP-MAC address of the device and login credentials must be matched with the previously stored login data and IP-MAC address. If there is any mismatching in both data, then access should be blocked by the Web amenity.

Any type of information or data can be stored in a computer through a database. A database is basically a collection of interconnected information about a person or organization. The database is basically found only in relational format, but many forms of database have appeared in modern times. Earlier Web amenities mostly used to store and process only one type of data, so only one type of database program was sufficient to handle the data. But the type of data has also changed a lot over time. Today, Web amenities have different hues of data available in different formats. Using just one database program to manage different hues of data will not be enough. Today, the demand of the times is that according to the fast changing format of data in Web amenities, it has become necessary to use various database programs to manage data in Web amenities.

Keeping these requirements in mind, a concept was born known as Polyglot Persistence [23]. This concept has its origins in polyglot programming, which mentions that in developing an application one should use different programming languages rather than using only one programming language. Keeping these requirements in mind, a concept was born known as Polyglot Persistence. This concept has its origins in polyglot programming, which mentions that in developing an application one should use different programming languages rather than using only one programming language. Similarly, when designing any database, one should use different hues of database programs available rather than using only one database program. Traditional RDBMS provides an expedient strategy to hoard and admittance data centered on a preordained edifice. RDBMS, as we know, was apprehended primarily in the milieu of transactional systems. Over time they have also been successful for rational applications and online analytical processing (OLAP). Along with all these topographies, RDBMS also has some inherent hitches. For example, the use of RDBMS to store and process data related to community topographies and digital-trade becomes enormously exorbitant and convoluted.

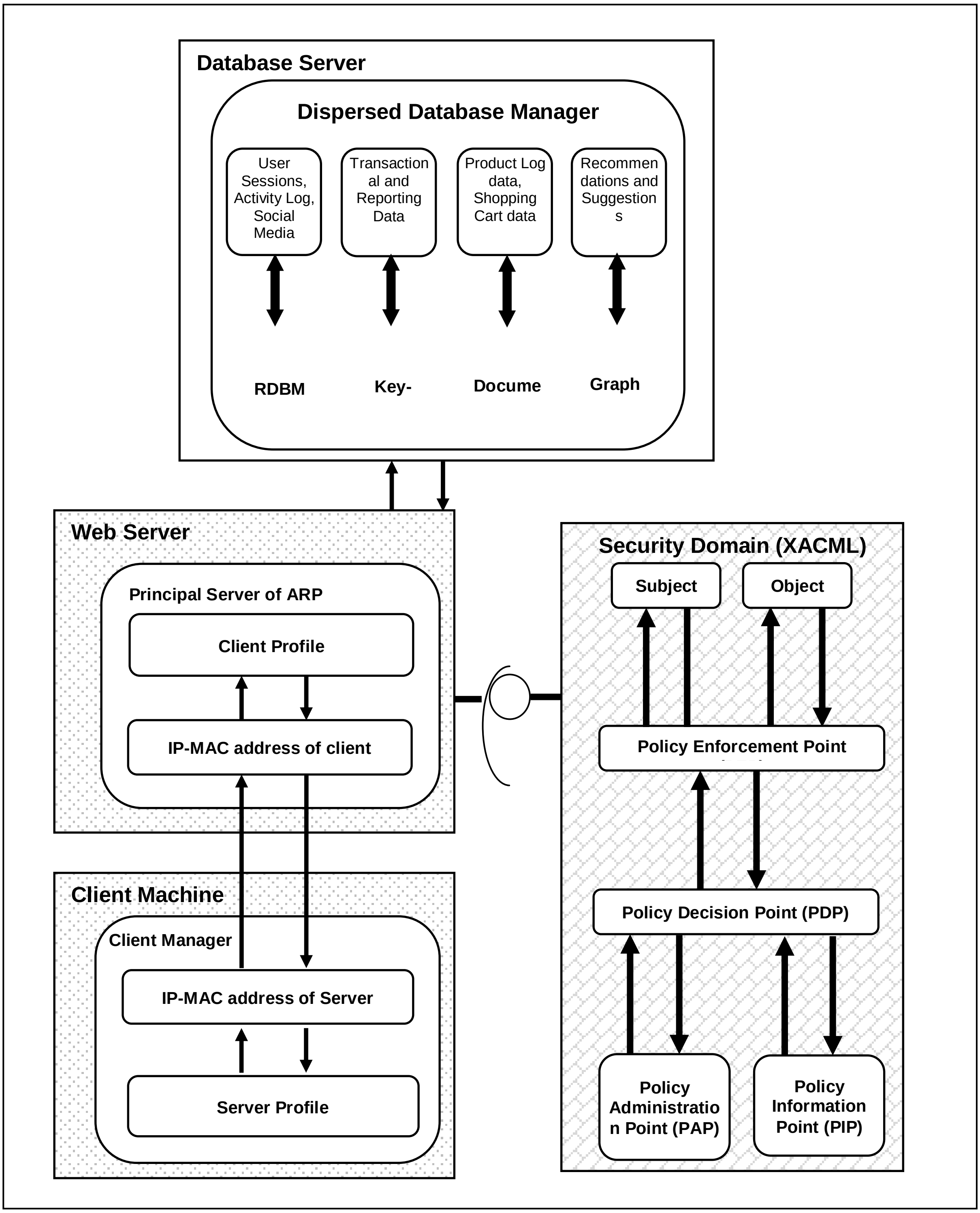
Database Server
Dispersed Database Manager
User Sessions, Activity Log, Social Media
Transactional and Reporting Data
Product Log data, Shopping Cart data
Recommendations and Suggestions
RDBM
Key-
Docume
Graph
Web Server
Principal Server of ARP
Client Profile
IP-MAC address of client
Client Machine
Client Manager
IP-MAC address of Server
Server Profile
Security Domain (XACML)
Subject
Object
Policy Enforcement Point
Policy Decision Point (PDP)
Policy Administration Point (PAP)
Policy Information Point (PIP)

**Figure 1: Integrated Threat Detection Framework using Dispersed Database Approach and IP-MAC Trussing for Secure Semantic Web**

Under these scenarios, an integrated threat prevention framework using a dispersed database ploy and IP-MAC trussing is insinuated. The insinuated framework integrates with XACML to enhance its capability. Fig. 1 shows the insinuated framework for securing the Semantic Web amenities against network assails such as Renunciation-of-Amenity (RoA) and Man-In-The-Middle (MITM) assail. The insinuated framework enriches the proficiency of XACML by introducing the concept of dispersed database ploy for managing assorted type of data available on digital-trade Web amenity and provision for preventing the Web amenities from network assails. The insinuated dispersed database ploy for amassing the Web amenity data such as digital-trade uses a set of database programs for amassing assorted type of data available on a digital-trade Web amenity. The framework suggests the use of dispersed database ploy for amassing diversified data that is attainable on digital-trade Web amenity.

The insinuated framework has three modules: Dispersed Database Manager, Principal Server of ARP and Client Manager. The Dispersed Database Manager is responsible for management dispersed database available on the digital-trade Web amenity. The Principal Server of ARP is responsible for maintaining the profile of registered client at Web server end. It hoards the logical and physical address of the client along with their personal detail such as user id, password, email, phone etc. The Client Manager is responsible for maintaining the server profile at client end. The combination of logical and physical address is denoted as IP-MAC trussing and this IP-MAC trussing will be used to prevent the Web server as well client from Renunciation-of-Amenity (RoA) and Man-In-The-Middle (MITM) assails. Generally, these assails are originated from the devices whose MAC address is always varies from the MAC address of registered client device. So the perception is to whenever the request and/or retort is originated from a new MAC address, the Web server and client should match the current IP-MAC trussing with registered IP-MAC trussing of both client and server. If any mismatch, the request and/or retort should be revoked from execution. Fig. 2 shows the flowchart of the Principal Server of ARP and client manager.

### 3.1. Dispersed Database Manager

The framework has dispersed database manager module that manages the unalike variations of data available on a digital-trade Web amenity. The module will organize the data in to different database programs and specifies which database suite will be used for amassing anunalike sort of data. During data access, the module will also help in suggesting the appropriate database programs for accessing a specific type of data. In a Web amenity such as digital-trade, a solitary database would not adequate to triumph assorted data requisite. Instead of depleting the alike database to endure multifarious sorts of data, it ought to be amended to hoard the data in clusters of unalike databases.

A digital-trade Web amenity will obligate to knob disparate discrepancies of data such as shopping dray and session data, purchaser order data and transaction data, tracking data, stockpile administration information, purchaser community association and their procurement and probing emulate information, artifact information, and reporting information etc. The dispersed database manager will helping in suggesting the use of a specific database program for managing such genre of diverse data. For example, it is better to hoard user session data and social media analysis data in a key value type of database in place hoarding it in RDBMS. Transactional and reporting data will give better result, if it will be hoard in a RDBMS. Likewise, Product log data and shopping cart data will readily be manage in document type database. Recommendations and suggestions will produce better outcome when gets stored in graph-type database. Thus, the module will aids in managing the assorted type of data.

### 3.2. Principal Server of ARP

The Principal server of ARP sub-module will maintain the profile of each client connected to the server. Principal server of ARP stores the IP and MAC address of registered clients along with their user id and password. Whenever a new client connects with the server, the Principal server of ARP will store the credentials of client along with logical and physical address of the client in its database. The Principal server of ARP will maintain the ancillary index, which stores the IP-MAC trussing of the client. This IP-MAC trussing will be used to authenticate the validity of the client. When a client generates a request to the Web server, the Principal server of ARP will check the IP-MAC trussing of the requesting client. The Principal server of ARP will compare the current IP-MAC trussing of the client with previously stored IP-MAC address of the client. If both IP-MAC addresses are same, the Principal server of ARP will authenticate the client.

If there is a mismatch in IP-MAC trussing of the client, the Principal server of ARP will send few packets to the previous MAC address of the client. If at least one reply sent by the previous MAC address, the new MAC address is discarded by the server. The Principal server of ARP will keep the previous IP-MAC trussing in its database server. If no reply is received from previous MAC address, the Principal server of ARP will store the new IP-MAC trussing and update the entry in ancillary index. This way it will revoke the Renunciation-of-Amenity (RoA) and Man-In-The-Middle (MITM) assails on the Web server.

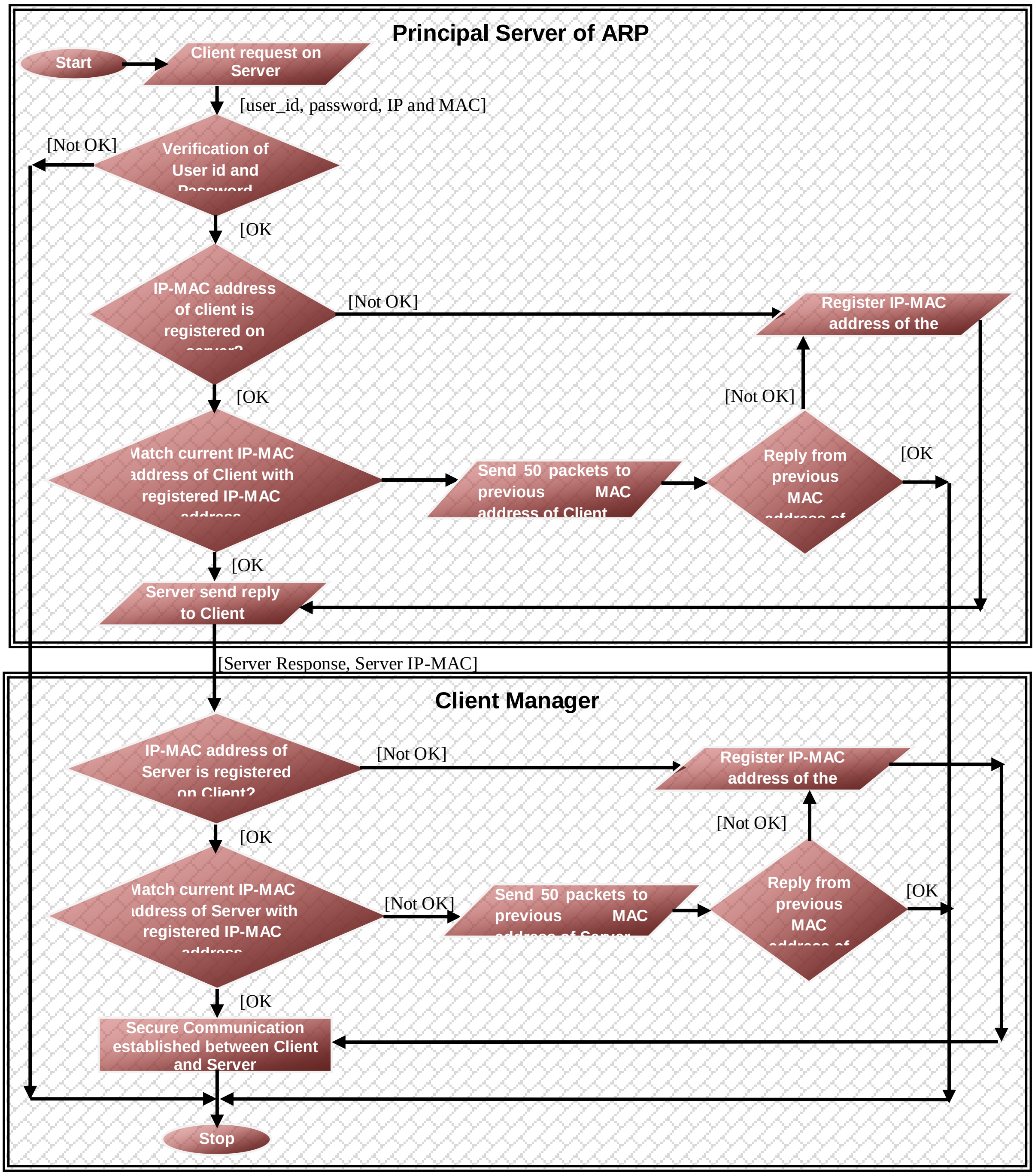
Principal Server of ARP
Start
Client request on Server
[user_id, password, IP and MAC]
[Not OK]
Verification of User id and Password
[OK
IP-MAC address of client is registered on server?
[Not OK]
Register IP-MAC address of the
[OK
[Not OK]
Match current IP-MAC address of Client with registered IP-MAC address
Send 50 packets to previous MAC address of Client
Reply from previous MAC address of
[OK
[OK
Server send reply to Client
[Server Response, Server IP-MAC]
Client Manager
IP-MAC address of Server is registered on Client?
[Not OK]
Register IP-MAC address of the
[OK
[Not OK]
Match current IP-MAC address of Server with registered IP-MAC address
[Not OK]
Send 50 packets to previous MAC address of Server
Reply from previous MAC address of
[OK
[OK
Secure Communication established between Client and Server
Stop

**Figure 2: Flowchart of Principal Server of ARP and Client Manager**

**3.3Client Manager**The client manager sub-module will responsible for maintaining the IP-MAC address of principal server of ARP. The client host maintains an ancillary ARP index that will permanently store the IP-MAC address of the principal server of ARP. Whenever a client wants to communicate with the Web server, its request is transferred to the principal server of ARP. The principal server of ARP transfers its IP-MAC address to the client to let the client know that the reply to its request iscoming from a legitimate Web server. The client preserves theIP-MAC address of the principal server of ARP in its ancillary index, so that the validity of the server can be identified next time. Whenever the client receives the retort from the Web server, the client manager will ensure that the IP-MAC trussing of the Web server is available in its database. If the IP-MAC trussing of the Web server is not already stored then the client manager will store the IP-MAC trussing of the Web server and accept the retort sent by the server. If the IP-MAC trussing of the Web server is stored with the client manager, it will match the current IP-MAC address to the previous IP-MAC trussing. If both IP-MACs are the same then the client manager will accept the retort sent by the Web server. But if there is a difference between the current IP-MAC trussing and the previous IP-MAC, the client manager will send few packets to the previously registered IP-MAC trussing. If a single retort is received from the previous MAC address, the client will discard the current MAC of the Web server and will not accept the retort. If there is no retort from the previous mac of the Web server, then the client manager will update the current IP-MAC trussing of the Web server in the database and will also accept the retort sent by the Web server. In this way the client manager will not accept the retort from any unique MAC address and will further enhance the security of the client.

## 4. Conclusion and Future Work

This paper insinuated the threat prevention framework for preventing the assail on the Semantic Web amenities based on XACML. The insinuated framework suggests the use of dispersed database technologies for amassing various unalike data available on digital-trade Web amenity. This framework also proposes the concept of IP-MAC trussing for both client and server. The database server known as principal server of ARP will store the IP-MAC trussing of each client registered on the server. In turn, the client will also store the IP-MAC trussing of registered server. When a registered client connects with the principal server of ARP, the principal server of ARP will match the IP-MAC trussing of the client with previously stored IP-MAC trussing. If both IP-MAC trussing are same, the client will be given access to use server resources. If there is a mismatch, the principal server of ARP will send few packets to previous MAC address of the client. If atleast one response is coming from the previous MAC, the current from new MAC address will be discarded by the server. The same process is followed by the client. This way, the insinuated framework will be able to tackle the network assails such as Renunciation-of-Amenity (RoA) and Man-In-The-Middle (MITM) assail.

In the future, the focus will be on further refining the insinuated framework for semantic Web amenities and the insinuated framework may also be applied to other domains of semantic web amenities.

## 5. REFERENCES

1. Badsha, S., Yi, X., Khalil, I., Liu, D., Nepal, S. and Lam, K.Y., 2018. Privacy preserving user based web amenity recommendations. IEEE Access, 6, pp.56647-56657.
2. Huang, K., Zhang, J., Tan, W., Feng, Z. and Chen, S., 2016. Optimizing semantic annotations for web amenity invocation. IEEE Transactions on Services Computing, 12(4), pp. 590-603.
3. Huang, Y., Chai, Y., Liu, Y. and Shen, J., 2018. Architecture of next-generation digital-trade platform. Tsinghua Science and Technology, 24(1), pp.18-29.
4. Li, X., Xue, Y. and Malin, B., 2012, October. Detecting anomalous user behaviors in workflow-driven web applications. In 2012 IEEE 31st Symposium on Reliable Distributed Systems (pp. 1-10). IEEE.
5. Amini, M. and Osanloo, F., 2016. Purpose-based privacy preserving access control for secure amenity provision and composition. IEEE Transactions on Services Computing 12(4), pp. 604-620.
6. Liu, A.X., Chen, F., Hwang, J. and Xie, T., 2010. Designing fast and scalable XACML policy evaluation engines. IEEE Transactions on Computers, 60(12), pp.1802-1817.
7. Karanikola, L. and Karali, I., 2018. Towards a Dempster–Shafer Fuzzy Description Logic—Handling Imprecision in the Semantic Web. IEEE Transactions on Fuzzy Systems, 26(5), pp.3016-3026.
8. Zhang, Q., DiFranzo, D., Gloria, M.J.K., Makni, B. and Hendler, J.A., 2018. Analyzing the Flow of Trust in the Virtual World With Semantic Web Technologies. IEEE Transactions on Computational Social Systems, 5(3), pp.807-815.
9. Liu, F., Deng, D., Jiang, J. and Tang, Q., 2018. Event-Driven Semantic Service Discovery Based on Word Embeddings. IEEE Access, 6, pp.61030-61038.
10. Li, J. and Lin, J., 2019. Research on the Influence of Sampling Methods for the Accuracy of Web Services QoS Prediction. IEEE Access, 7, pp.39990-39999.
11. Ye, H., Cao, B., Peng, Z., Chen, T., Wen, Y. and Liu, J., 2019. Web Services Classification Based on Wide & Bi-LSTM Model. IEEE Access, 7, pp.43697-43706.
12. Ranchal, R., Bhargava, B., Angin, P. and Othmane, L.B., 2018. Epics: A framework for enforcing security policies in composite web amenities. IEEE Transactions on Services Computing.
13. Diez, F.P., Vasu, A.C., Touceda, D.S. and Cámara, J.M.S., 2017. Modeling xacml security policies using graph databases. IT Professional, 19(6), pp.52-57.

**14.** Lonetti, F. and Marchetti, E., 2018. On-line tracing of XACML-based policy coverage criteria. IET Software, 12(6), pp.480-488.

**15.** Ilhan, O.M., Thatmann, D. and Kupper, A., 2015, November. A performance analysis of the XACML decision process and the impact of caching. In 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 216-223). IEEE.

**16.** Mourad, A. and Jebbaoui, H., 2014, July. Towards efficient evaluation of XACML policies. In 2014 Twelfth Annual International Conference on Privacy, Security and Trust (pp. 164-171). IEEE.

**17.** Atiq, A.M. and Alsulaiman, L.A., 2016, March. Using XACML to enhance compliance with privacy regulations in health sector. In 2016 World Symposium on Computer Applications & Research (WSCAR) (pp. 53-58). IEEE.

**18.** Castrejón, J., Vargas-Solar, G., Collet, C. and Lozano, R., 2013, September. ExSchema: Discovering and maintaining schemas from polyglot persistence applications. In 2013 IEEE International Conference on Software Maintenance (pp. 496-499). IEEE.

**19.** Sellami, R. and Defude, B., 2017. Complex queries optimization and evaluation over relational and NoSQL data stores in cloud environments. IEEE Transactions on Big Data, 4(2), pp.217-230.

**20.** Tan, S.Y., 2018. Comment on "Secure Data Access Control With Ciphertext Update and Computation Outsourcing in Fog Computing for Internet of Things". IEEE Access, 6, pp.22464-22465.

**21.** Liu, Q., Zhang, H., Wan, J. and Chen, X., 2017. An access control model for resource sharing based on the Role-Based access control intended for Multi-domain manufacturing Internet of Things. IEEE Access, 5, pp.7001-7011.

**22.** Ammar, N., Malik, Z., Bertino, E. and Rezgui, A., 2015. XACML policy evaluation with dynamic context handling. IEEE Transactions on Knowledge and Data Engineering, 27(9), pp.2575-2588.

**23.** G. Vial, "Different Databases for Different Strokes", IEEE Software, vol. 35, Mar. 2018 pp. 80-85.