International Journal of Research in Engineering, Science and Management
Volume 5, Issue 1, January 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Designing Technique for Protecting the Composition Results from Privacy Attacks

Kachare Jitendra Vilas[1], Chavan Vijatsingh[2], Jadhav Hiraman Sheshrao[3*], Omkar Kalyan Wadne[4]
[1]*Student, Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur, India*
[2]*Professor, Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur, India*
[3,4]*Lecturer, Zeal Polytechnic, Pune, India*

***Abstract*: Data as a Service (DaaS) builds on service-oriented technologies to enable fast access to data resources on the Web. However, this paradigm raises several new privacy concerns that traditional privacy models do not handle. In addition, DaaS composition may reveal privacy-sensitive information. In this paper, I am proposing a formal privacy model in order to extend DaaS descriptions with privacy capabilities. The privacy model allows a service to define a privacy policy and a set of privacy requirements. I am also proposing a privacy-preserving DaaS composition approach allowing to verify the compatibility between privacy requirements and policies in DaaS composition. I was proposing a negotiation mechanism that makes it possible to dynamically reconcile the privacy capabilities of services when incompatibilities arise in a composition. I validate the applicability of my proposal through a prototype implementation and a set of experiments.**

***Keywords*: attacks, designing, technique, protecting, privacy.**

1. Introduction

Web services have recently emerged as a popular medium for data publishing and sharing on the Web. Modern enterprises across all spectra are moving towards a service-oriented architecture by putting their databases behind Web services, thereby providing a well-documented, platform independent and interoperable method of interacting with their data. This new type of services is known as DaaS (Data-as-a-Service) services, where services correspond to calls over the data sources. DaaS sits between services-based applications (i.e. SOA-based business process) and an enterprise’s heterogeneous data sources. They shield applications developers from having to directly interact with the various data sources that give access to business objects, thus enabling them to focus on the business logic only. While individual services may provide interesting information/functionality alone, in most cases, users’ queries require the combination of several Web services through service composition. In spite of the large body of research devoted to service composition over the last years, service composition remains a challenging task in particular regarding privacy. In a nutshell, privacy is the right of an entity to determine when, how, and to what extent it will release private information. Privacy relates to numerous domains of life and has raised particular concerns in the medical field, where personal data, increasingly being released for research, can be or have been, subject to several abuses, compromising the privacy of individuals.

2. Background

The approach presented in this paper is implemented as a part of which deals with the privacy preservation issue in P2P data sharing environments, particularly in epidemiological research where the need of data sharing is apparent for making better a health environment of people. To support the decision process, epidemiological researchers should consider multiple data sources such as the patient data, his social conditions, the geographical factors, etc. The data sources are provided by DaaS services and are organized with peers. DaaS services differ from traditional Web services, in that they are stateless; i.e., they only provide information about the current state of the world but do not change that state. When such a service is executed, it accepts from a user an input data of a specified format ‘‘typed data’’ and returns back to the user some information as an output. DaaS services are modeled by RDF views. With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

Despite important regulatory and technical efforts aimed at preserving privacy, privacy leakage incidents on the Web continue to make the headlines. We present a formal model for preserving privacy in Web services. We define a Web service-aware privacy model that deals with the privacy of input data, output data, and operation usage. We introduce a matching protocol that caters for partial and total privacy compatibility. We propose also a negotiation model to reconcile services' privacy in case of incompatibility.

*Corresponding author: hiraman.jadhav123@gmail.com

3. Proposed System

Web service composition enables seamless and dynamic integration of business applications on the web. The performance of the composed application is determined by the performance of the involved web services. Therefore, non-functional, quality of service aspects is crucial for selecting the web services to take part in the composition. Identifying the best candidate web services from a set of functionally-equivalent services is a multi-criteria decision making problem. The selected services should optimize the overall QoS of the composed application, while satisfying all the constraints specified by the client on individual QoS parameters. In this paper, we propose an approach based on the notion of skyline to effectively and efficiently select services for composition, reducing the number of candidate services to be considered. We also discuss how a provider can improve its service to become more competitive and increase its potential of being included in composite applications. We evaluate our approach experimentally using both real and synthetically generated datasets.

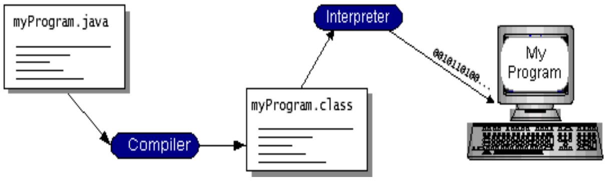

Fig. 1. Java programing language

This present disclosure relates to the novel work carried for implementation of a Hidden Owner Identity mechanism for securely storing file on storage server. Moreover, it relates to unique, efficient and secure method for storing a file on any type of storage server either local or cloud-based server. Furthermore, present disclosure relates to computer program product consisting of computer -readable instructions stored on computer -readable storage media. These instructions being executed by computing system consisting of processing hardware to execute programs. File storage server is a server which stores various types of critical user data files and privacy -sensitive information using file systems therefore they are main targets for various types of security attacks.

4. Implementation

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works of client to start file processing over the developed environment. After getting all the credentials for file processing client is allow to start working over the environment. Client select a file to be uploaded and generate a secure key to make a request to upload a file into the available cloud. The next step calls the Encryption and splitting API's to perform the secure distribution of file over the cloud. The timeline process call also be called in reverse to get file receive request made satisfaction of the client.

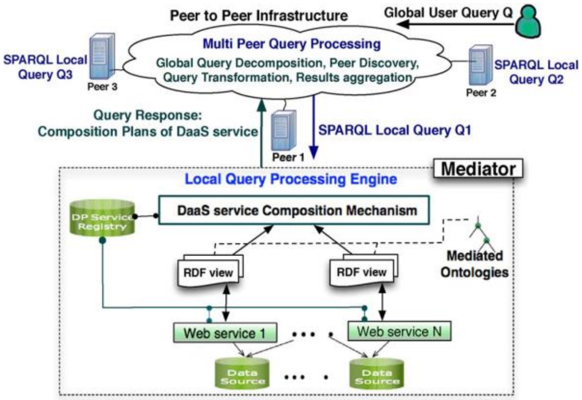
Peer to Peer Infrastructure
Global User Query Q
Multi Peer Query Processing
Global Query Decomposition, Peer Discovery, Query Transformation, Results aggregation
SPARQL Local Query Q3
Peer 3
SPARQL Local Query Q2
Peer 2
Query Response: Composition Plans of DaaS service
Peer 1
SPARQL Local Query Q1
Mediator
Local Query Processing Engine
DP Service Registry
DaaS service Composition Mechanism
RDF view
RDF view
Mediated Ontologies
Web service 1
Web service N
Data Source
Data Source

Fig. 2. Web services

The Java Programming Language:

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

5. Conclusion

In this project, we proposed a dynamic privacy model for Web services. The model deals with privacy at the data and operation levels. We also proposed a negotiation approach to tackle the incompatibilities between privacy policies and requirements. Although privacy cannot be carelessly negotiated as typical data, it is still possible to negotiate a part of privacy policy for specific purposes. In any case, privacy policies always reflect the usage of private data as specified or agreed upon by service providers. As a future work, we aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator.

References

[1] M. Alrifai, D. Skoutas, and T. Risse, "Selecting Skyline Services for QoS-Based Web Service Composition," in Proc. 19th Int'l Conf., 2010, pp. 11-20.

[2] M. Barhamgi, D. Benslimane, and B. Medjahed, "A QueryRewriting Approach for Web Service Composition," IEEE Trans.Serv. Comput., vol. 3, no. 3, pp. 206-222, July-Sept. 2010.

[3] G.T. Duncan, T.B. Jabine, and V.A. de Wolf, “Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics,” Washington, DC, USA: Nat. Acad. Press, 1993.

[4] B.C.M. Fung, T. Trojer, P.C.K. Hung, L. Xiong, K. Al-Hussaeni, and R. Dssouli, “Service-oriented Architecture for High-Dimensional Private Data Mashup,” IEEE Trans. Serv. Comput.,

[5] Y. Gil, W. Cheung, V. Ratnakar, and K. K. Chan, “Privacy Enforcement in Data Analysis Workflows,” in Proc. Workshop PEAS ISWC/ASWC, vol. 320, CEUR Workshop Proceedings, T. Finin, L. Kagal, and D. Olmedilla, Eds., Busan, South Korea, Nov. 2007, CEUR-WS.org.

[6] A. H. H. Ngu, M. P. Carlson, Q. Z. Sheng, and H. Y. Paik, “Semantic-Based Mashup of Composite Applications,” IEEE Trans. Serv. Comput., vol. 3, no. 1, pp. 2-15, Jan.-Mar 2010.

[7] N. Mohammed, B.C.M. Fung, K. Wang, and P.C.K. Hung, “Privacy-Preserving Data Mashup,” in Proc. 12th Int’l Conf. EDBT, 2009, pp. 228-239.

[8] Y. Lee, J. Werner, and J. Sztipanovits, “Integration and Verification of Privacy Policies Using DSML’s Structural Semantics in a SOA-Based Workflow Environment,” J. Korean Soc. Internet Inf., vol. 10, no. 149, pp. 139-149, Aug. 2009.

[9] Y. Lee, D. Sarangi, O. Kwon, and M.-Y. Kim, “Lattice Based Privacy Negotiation Rule Generation for Context-Aware Service,” in Proc. 6th Int’l Conf. UIC, 2009, pp. 340-352.