A New Privacy Preserving Access Control Policy for Event Processing System

K REVATHI PRASANNA [#1], CH SRINIVASA REDDY[*2], P SRINIVAS[*3]

[#1]M.TECH, Department of information technology, vignan's institute of Information Technology, JNTU-KAKINADA, Andhra Pradesh, India
[#1]prasanna.k231@gmail.com

[*2]Assistant professor, Department of information technology, vignan's institute of Information Technology, Visakhapatnam, Andhra Pradesh, India
[*2] Srinivasreddyviit@gmail.com

[*3]Assistant Professor, Department of information Technology, Vignan's Institute of Information Technology, Visakhapatnam, Andhra Pradesh, India
[*3] Srinivasp3@gmail.com

Abstract

Event processing is a method of tracking and analyzing (processing) streams of information (data) about things that happen (events) and deriving a conclusion from them. Complex event processing, or CEP, is event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. Today event processing systems lack methods to preserve privacy constraints of incoming event streams in a chain of subsequently applied stream operations. This problem is mainly observed in large-scale distributed applications like a logistic chain where there were a lot of individual domains available for processing the event. An intruder can always infer from legally received outgoing event streams confidential input streams of the event processing system. This paper mainly concentrates a very new fine-grained access management for processing complex event processing tasks. Here each and every operation is performed by individual roles with their access policies specified. This paper is mainly used for specifying the access policy and also enforcement of those access policy specifications in a proper way. By conducting various experiments on our proposed access policy system, we finally came to a conclusion that this access control policy clearly suits for almost all types of logistics for performing their operations without any misuse in transaction. By conducting various experiments on real time courier/shipping company web sites, we finally came to an conclusion that the current application suits best for avoiding fake during courier deliveries.

Keywords

Event Processing System, Security, Access control System, Fine Grained Access Control Policy, Shipping Company

1. Introduction

Event processing is a method of tracking and analyzing (processing) streams of information (data) about things that happen (events) and deriving a conclusion from them. Complex event processing, or CEP, is event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. Today event processing systems lack methods to preserve privacy constraints of incoming event streams in a chain of subsequently applied stream operations. In large scale organization or business processes, it is essential to detect inconsistencies or failures early. For example, in large scale manufacturing and logistic chain processes, transaction of items are tracked continuously to detect loss or to re-route them during transport. To answer this need complex event processing (CEP) systems have evolved as a key paradigm for business and industrial applications [1], [2]. CEP systems allow to detect situations by performing operations on event streams which emerge from sensors all over the world, e.g. from packet tracking devices.

While traditionally event processing systems have applied powerful operators in a central way, the emerging increase of event sources and event consumers have raised the need to reduce the communication load by distributed in-network processing of stream operations [3], [4], [5], [6]. In addition, the collaborative nature of today's economy results in large-scale networks, where different users, companies, or groups exchange events. As a result, event processing networks are heterogeneous in terms of processing capabilities and technologies, consist of differing participants, and are spread across multiple security domains [7], [8]. However, the increasing interoperability of CEP applications raises the question of security [2]. It is not feasible for a central instance to manage access control for the whole network. Instead, every producer of information should be able to control how its produced data can be accessed. For example, a company may restrict certain information to a subset of authorized users (i.e. that are registered in its domain). Current work in providing security for event-based systems covers already confidentiality of individual event streams and the authorization of network participants [9],[10], [11]. In CEP systems, however, the provider of an event looses control on the distribution of *dependent* event streams. This constitutes a major security problem, allowing an adversary to infer information on confidential ingoing event streams of the CEP system.

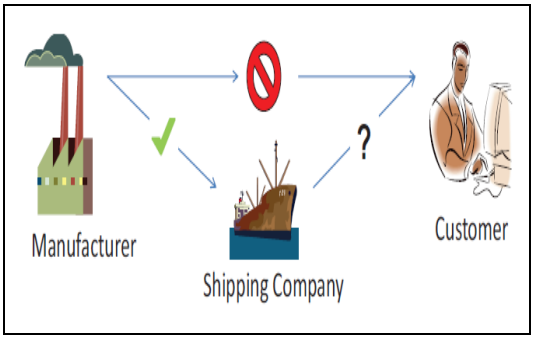


Figure 1. Access Control & Event Dependency

For Example consider the following logistic shipping process illustrated in Figure 1 where a manufacturer is a role who mainly inserts the products in his warehouse and he wants to deliver an item to a customer. The shipping company is mainly used for determining a warehouse which is very close to the customer who ordered the product, where the item will be shipped to that warehouse before it will be delivered to the customer. The entire logistic process is supported by an event processing system, where operators are hosted in the domain of each party and exchange events including potentially confidential information (e.g. the item's *destination* is transmitted to the shipping company). If now a third party receives events related to the *warehouse*, it may draw conclusions about the original event data (i.e. *destination address*), in spite of the manufacturer declaring this information as highly confidential and only providing the shipping company with access rights to it.

The main goal of this present event processing system is to establish access control that ensures the privacy of information even over multiple processing steps in a multi-domain, large scale CEP system. In particular, our contributions for this paper are i) an *access policy inheritance* mechanism to enforce access policies over a chain of dependent operators and ii) a scalable method to measure the *obfuscation* imposed by operators on information exchanged in event streams. This allows to define as part of the *access policy* an obfuscation threshold to indicate when the event processing systems can ignore access restrictions, thus increasing the number of events to which application components can react to and this way increasing also the utility of the CEP system.

2. Literature Survey

In this section we will describe the assumptions that are used in the proposed paper. This section mainly surveys on the literature of our proposed new privacy preserving access control policy for event processing systems. Here we mainly discuss about the logistics what we are using for event processing system.

2.1 Logistics

Logistics is the management of the flow of goods between the point of origin and the point of consumption in order to meet some requirements, of customers or corporations. The resources managed in logistics can include physical items, such as food, materials, animals, equipment and liquids, as well as abstract items, such as time, information, particles, and energy. The logistics of physical items usually involves the integration of information flow, material handling, production, packaging, inventory, transportation, warehousing, and often security. The complexity of logistics can be modeled, analyzed, visualized, and optimized by dedicated simulation software. The minimization of the use of resources is a common motivation in logistics for import and export which is clearly shown in figure 2.

Figure 2. Represents a Logistic Warehouse

2.2 Main Motivation

We assume a distributed correlation network, where dedicated hosts are interconnected. On these hosts we deploy operators, which are executed to collaboratively detect situations and form the distributed CEP system. The cooperative behavior of the operators is modeled by a directed operator graph $G = (\Omega, S)$ which consists of operators $\omega \in \Omega$ and *event streams* $(\omega i, \omega j) \in S \subseteq (\Omega \times \Omega)$ directed from ωi to ωj . Thus, we call ωi the event *producer* and ωj the *consumer* of these events. Each event contains one or more event attributes which have discrete values. Every operator ω implements a correlation function $f\omega : I\omega \rightarrow O\omega$ that maps incoming event streams $I\omega$ to outgoing event streams $O\omega$. In particular, $f\omega$ identifies which events of its incoming streams are selected, how event patterns are identified (correlated) between events, and finally how events for its outgoing streams are produced.

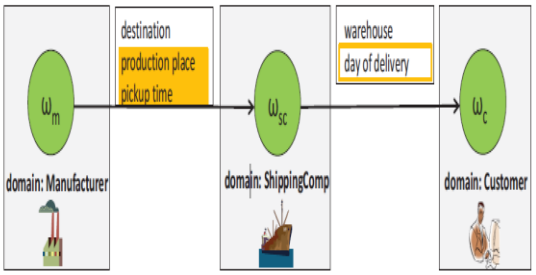


Figure 3. Represents the attributes in Shipping Scenario

Figure 3 illustrates clearly an operator graph of three operators according to the introduced logistics example, each operator hosted in a distinct domain. The correlation function *fsc* is applied to events received from and produced by ωm on produced items in the manufacturing domain. Events produced by *fsc* carry two *event attributes*, the warehouse location and estimated day of delivery for shipped items.

3. Goals of Our Proposed Access Control Model

The following are the goals of our proposed access control policy which was used in large scale logistics for doing a chain of online transactions.

3.1 Access Control for CEP

Our approach allows inheriting access requirements by assigning them to event attributes in form of an *access policy*. This allows to preserve requirements through any chain of dependent correlation steps of operators in *G*. In addition, an obfuscation policy allows to specify an *obfuscation threshold* for event attributes. In each correlation step, the obfuscation of event attributes in produced events is determined by the proposed access policy consolidation protocol. Once the obfuscation threshold is reached for an event attribute, the attribute's access requirements can be ignored. In the following, we detail the concepts behind access policies and obfuscation policies, and formalize the security goal.

3.2 Access Policies

Access control allows specifying access rights of subjects (operators) for the set of available objects (event attributes). These access rights are provided by the owner of an object (e.g. the producer of an event stream) and may be granted to operators based on an *access requirement*. Such a requirement may be a role, a location or a domain affiliation. Requirements are usually not direct *properties* of the operators, but of the hosts where the operators are deployed. Formally, we specify the access rights within an *access policy AP* for an operator ω as a set of (attribute, access requirement) pairs:

$$AP\omega = \{(att1, ar1), ..., (attn, arn)\} .$$

If there is no requirement specified for an attribute, any consumer in the network will be able to access it. Note that we consider attributes to be distinct even if they use the same name, but are produced at two distinct operators. An access requirement is a tuple of a property *p*, a mathematical operator *op* and a value set *val*: *ar* = (*p*, *op*, *val*),

Where $op \in \{=,<,>,\leq,\geq, \in\}$. ***val*** can be specified by a range or a set of values. For the sake of simplicity, in this paper access requirements are only referring to domain affiliation and have a structure like this:

$$ar1 = (domain, \in, \{domainA, domainB\}).$$

In our example scenario, the manufacturer's event attributes have different access requirements. While the information about the item's destination is accessible by the customer, information about where the item is produced and when it can be picked up is restricted to the shipping company.

Therefore, the attached AP is defined as follows:

APmanufacturer =

{(destination,(domain,∈,{shippingComp,customer}}),

(pickup time, (domain,=,shippingComp)),

(production place, (domain,=,shippingComp))}

With the enforcement and assurance of access policies at each producer, a consumer will be eligible to access (receive) an attribute only if the consumer's properties match the access requirements defined for the particular attribute. In this case the consumer is trusted to use the attribute in its correlation function and adopt the requirements specified for the attribute in its own access policy for all produced events.

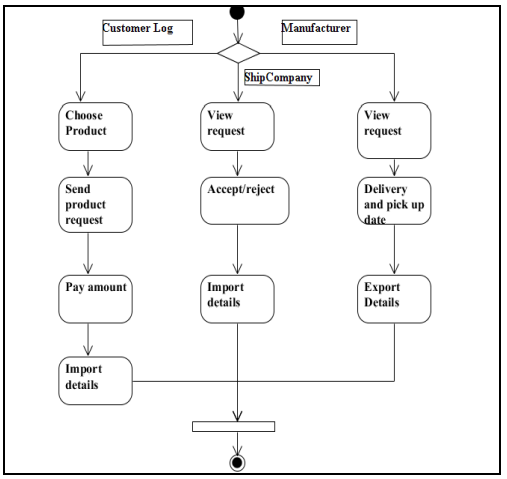


Figure 4. Represents the Architecture flow of our proposed access control policy in Shipping Scenario

4. Methodology of Proposed Paper

In this section we will discuss about the proposed algorithm that was used in order to prevent the miss-use of access policies in shipping scenario. For this we use **Local Obfuscation Calculation** algorithm in order to prevent the un-wanted access of intruder between customer and manufacturer as well as manufacturer and shipping company.

The below algorithm is mainly used for identifying the changes that was happened in the address field of the customer who book the item in online. Here the change was identified with two attributes called as old and new where the old attribute is nothing but the address filed that was mentioned in the records while doing online transaction initially. If the address was changed by any of the intruder or any third party warehouse location during the product transmission, it will be identified as new attribute ,through which we can identify that there was a difference in old and new attributes and we can able to stop the product transmission at this stage.

Step by Step Procedure

Procedure INITIALIZE(ω)

for all operator ω do
$D\omega \leftarrow$ FINDMULTIPATHOPERATORS(ω)
end for
for all $\omega \in D\omega$ do
relAtts ←FINDRELATEDATTRIBUTES
for all (*attnew*, *attold*) $\in$ *relAtts* do
TRANSMIT P(*attnew*|*attold*) TO ω
end for
end for
end procedure

Procedure UPONRECEIVEEVENT (*e*)

for all *att* $\in e$ do
if $\exists$ multPathDependency(att) then
CALCULATEWORSTCASEOBFUSCATION (ATT)
else
CALCULATELOCALOBFUSCATION(ATT)
end if
end for
end procedure

Any policy consolidation algorithm two conditions to be met:

Condition 1. *For all attributes att* $\in O\omega$ *produced at* ω *ARinit*(*att*) $\subset AP\omega$. (1)

Condition 2. *For all dependent attribute pairs* (*attold, attnew*) $\in \longrightarrow *$ *with*

1) ωi *has produced attold with access requirement AR*(*attold*) *and obfuscation threshold* (*attold, x*) $\in OP\omega i$,
2) *attnew is produced by* ωj
3) *attnew is consumed by* ωk *the access requirement in* $AP\omega j$ *yield*
AR(*attold*) $\subset AP\omega j$ *if obf* (*attold, attnew,* ωk) < *x.* (2)

5. Implementation Modules

Implementation is the stage where the theoretical design is automatically converted into practically by dividing this into various modules. We have implemented the current application in Java Programming language with Front End as java Swings, and Back End as SQL Server 2000 data base.

5.1 Main Modules

Our proposed application is divided into following 4 modules. They are as follows:

1) **Event Processing**
2) **Manufacturer**
3) **Shipping Company**
4) **Customer**

1) Event Processing Module

Event processing systems respond to events in the system's environment or user interface. The key characteristic of event processing systems is that the timing of events is unpredictable and the system must be able to cope with these events when they occur.

2) Manufacturer Module

In this module manufacturer, insert the product details and also view product request from shipping company. Send details to shipping company to delivery date and pickup time.

3) Shipping Company Module

In this module ship company, view product request from customer. Then company forward the request to manufacturer or reject the request.

4) Customer Module

In this module customer, product order from Ship Company and also views the order from Ship Company. Customer views the import details.

5.2 Main Source Code for Current Application

Here we will discuss the main source code for the proposed application implementation. Here I will show the data base logic through which the current application is connecting its frond end user interface with the storage data. For this application we are using JSP as front end and My Sql as back end data base.

Sample Code for Databasecon

```
package databaseconnection;
import java.sql.*;

public class databasecon
{
        static Connection con;
        public static Connection getconnection()
        {       try
                {

        Class.forName("com.mysql.jdbc.Driver");
                                con =
DriverManager.getConnection("jdbc:mysql://localho
st:3306/domain","root","admin");
                }
                catch(Exception e)
```



```
                {
        System.out.println("class error");
                }
                return con;
        }
}
```

Sample Code for Product_reg1

```
<%@ page import="java.sql.*,databaseconnection.*"%>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<title>Access Policy Consolidation for Event Processing Systems</title>

<meta name="keywords" content="" />

<meta name="description" content="" />

<link href="default.css" rel="stylesheet" type="text/css" media="screen" />

<script language="javascript" type="text/javascript" src="datetimepicker.js">

//Date Time Picker script- by TengYong Ng of http://www.rainforestnet.com

//Script featured on JavaScript Kit (http://www.javascriptkit.com)

//For this script, visit http://www.javascriptkit.com
</script>

</head>

<body bgcolor="#FFFFCC">
<!-- start header -->
<div id="header">
<font size="+3" color="#FF3333"><b>Access Policy Consolidation for Event Processing Systems</b></font> <p></p>
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu">
        <ul></ul>
</div>
<!-- end menu -->
<!-- start page -->
<div id="page">
        <!-- start content -->
        <div id="content">
        <%
                String name=(String)session.getAttribute("name");
                %>
                <p><small></small>Welcome: <%=name%>!</p>
<h2>Product Request Details</h2>
 <%
 String a=null,b=null,c=null,d=null,e=null,f=null,g=null,h=null;
    String mail=request.getQueryString();
        // out.print(mail);
//String fr=(String)session.getAttribute("fr");
try
{
Connection con = databasecon.getconnection();
PreparedStatement  ps=con.prepareStatement("select product,model,noofproduct,totalamt,destination,dates,company,place from orders where id='"+mail+"'");
ResultSet rs=ps.executeQuery();
if(rs.next())
{
a=rs.getString(1);
b=rs.getString(2);
c=rs.getString(3);
d=rs.getString(4);
e=rs.getString(5);
f=rs.getString(6);
g=rs.getString(7);
h=rs.getString(8);
%>

<form name="s" action="product_req2.jsp" method="get" onSubmit="return valid()">
<table>
<tr>
<td>
    <font color="#000000"><b>Company:</b></font><br>

    
<font color="#000000">
```

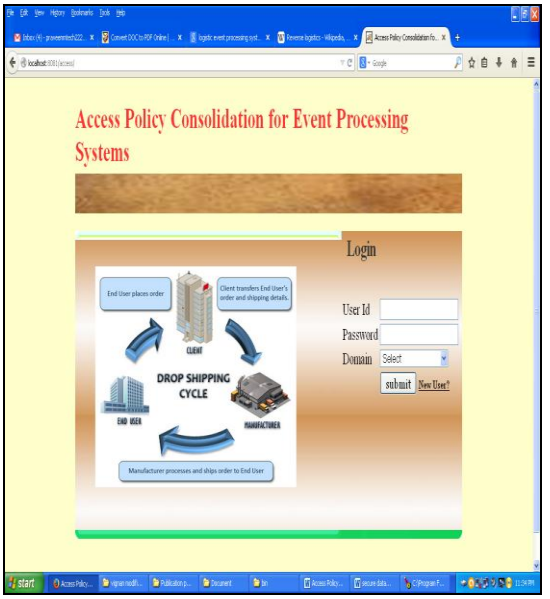


```
<b>Pick Up Place:</b></font><br>
    
<font
color="#000000"><b>Destination:</b></font><br>
    <font
color="#000000"><b>Product:</b></font><br>
    <font
color="#000000"><b>Model:</b></font><br>
    <font
color="#000000"><b>No                    Of
Product:</b></font><br>
    <font
color="#000000"><b>Paid
Amount:</b></font><br>
    <font
color="#000000"><b>Date                  Of
Booking:</b></font><br>
</td>
<td>
<input type="Text" name="comp"
value="<%=g%>" readonly="">
<br>
<input type="Text" name="pickup"
value="<%=h%>" readonly="">
<br>
<input type="Text" name="dest" value="<%=e%>"
readonly=""><br>
<input type="Text" name="product"
value="<%=a%>" readonly="">
<br>
<input type="Text" name="model"
</td></tr>
</table>
<%
}
catch(Exception e1)
{
out.println(e1.getMessage());
}
%>
```

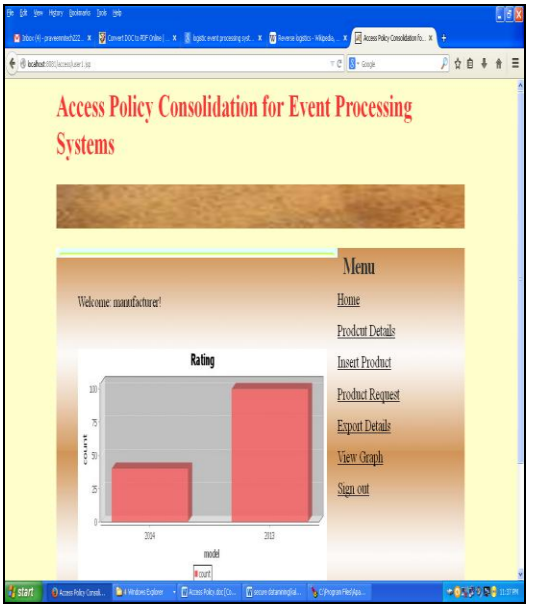
6. Experimental Results

In this paper, we have proposed a new secure algorithm called as **Local Obfuscation Calculation** Algorithm for mining horizontally distributed data bases. For example we took a logistic shipping company as an simulation environment, through which we showed the advantages of event processing system. In the logistic event processing system, there was mainly three actors available like Manufacturer, Shipping company, Customer, where each and every user has their individual roles and privileges. By designing this application we gave a new scope for the present logistics companies in order for processing there large complex events under the association of multiple users.We have shown the application in a web interface with JEE 6.0 edition .In this JEE we are using front end as Java Server pages (JSP) and HTML pages. As we are deploying the application in web interface, we are using tomcat server for deploying the application .Hence we use tomcat 7.0 as the deployment web server. We are using My Sql data base for storing the data temporarily on to our system and then retrieve the same data whenever needed. This MY-SQL data base is chosen on highest priority as it has GUI support and it is always Auto Commit by its nature.

Sample Screens



Represents Welcome page of the proposed event processing system



Represents Manufacturer Login page of the Proposed Event Processing System

7. Conclusion

In this paper, we have proposed an new multi-hop event processing network which is used for filing the gap that is available in the current event processing systems like shipping scenario. This paper is used mainly for addressing the inheritance and consolidation of access policies in heterogeneous CEP systems. We identified a lack of security in multi-hop event processing networks and proposed a solution to avoid this problem. Our algorithm includes the obfuscation of information, which can happen during the correlation process, and uses the obfuscation value as a decision-making basis whether inheritance is needed.

8. Future Enhancement

In the future we can extend the research of giving privacy preserving for event processing system in almost all types of domains what we choose for publishing. The proposed concept is very good in providing access control for event processing systems.

9. References

[1] A. Buchmann and B. Koldehofe, "Complex event processing," *it - Information Technology*, vol. 51:5, pp. 241–242, 2009.

[2] A. Hinze, K. Sachs, and A. Buchmann, "Event-based applications and enabling technologies," in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '09. New York, NY, USA: ACM, 2009, pp. 1:1–1:15.

[3] P. Pietzuch, "Hermes: A scalable event-based middleware," Ph.D. dissertation, University of Cambridge, 2004.

[4] G. Li and H.-A. Jacobsen, "Composite subscriptions in content-based publish/subscribe systems," in *Proc of the 6th Int. Middleware Conf.*, 2005, pp. 249–269.

[5] G. G. Koch, B. Koldehofe, and K. Rothermel, "Cordies: expressive event correlation in distributed systems," in *Proc. of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2010, pp. 26–37.

[6] B. Koldehofe, B. Ottenw¨alder, K. Rothermel, and U. Ramachandran, "Moving range queries in distributed complex event processing," in *Proc. of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2012, pp. 201–212.

[7] B. Schilling, B. Koldehofe, U. Pletat, and K. Rothermel, "Distributed heterogeneous event processing: Enhancing scalability and interoperability of CEP in an industrial context," in *Proc. of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2010, pp. 150–159.

[8] B. Schilling, B. Koldehofe, and K. Rothermel, "Efficient and distributed rule placement in heavy constraint-driven event systems," in *Proc. of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC)*, 2011, pp. 355–364.

[9] M. A. Tariq, B. Koldehofe, A. Altaweel, and K. Rothermel, "Providing basic security mechanisms in broker-less publish/ subscribe systems," in *Proceedings of the 4th ACM Int. Conf. on Distributed Event-Based Systems (DEBS)*, 2010, pp.38–49.

[10] L. I. W. Pesonen, D. M. Eyers, and J. Bacon,"Encryption-enforced access control in dynamic multidomain publish/subscribe networks," in *Proc. of the 2007 ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2007, pp. 104–115.

[11] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch, "Access control in publish/subscribe systems," in *Proc. of the 2nd ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2008, pp. 23–34.

