A CEP Privacy Security Access Control Framework Based on Event Attribute Detecting Tree

Bo Hong*
School of Computer Science and Engineering,
Xi’an Technological University,
Xi’an, 710021, China
e-mail: 1964383053@qq.com

Xin Jing
School of Computer Science and Engineering,
Xi’an Technological University,
Xi’an, 710021, China

***Abstract*—Complex event processing (CEP) technology is a study focus in the data flow processing area, while privacy security protection is the key problem that needs to be solved. In order to prevent illegal users from acquiring any information via registered event patterns, this paper discusses the CEP privacy security access control object in depth, formally defines four types of event attribute operators including completely read, partially read, access denied and quantity statistics, presents a privacy security protection engine with the event attribute detecting tree as the operating mechanism and puts forward a new feasible CEP privacy security access control framework based on this. The experimental result shows that such framework is able to realize efficient privacy information filtration based on the user role to reach the goal of CEP detecting information processing in a safe manner.**

Keywords-Complex Event Processing; Privacy Security Access Control; Event Attribute Detecting Tree; Event Attribute Operator; Security Protection Engine

I. INTRODUCTION

Data flow processing is a very important and active area in modern database technology. CEP technology[1] has become the study focus of such field since its inception as it is capable of integrating the information from the numerous data source distributed and digging the valuable dynamic meaning among the information from the high-speed data flow in real time. CEP technology is thoroughly changing the way of subscription & distribution and application data of the traditional information system. It acts as the hub of information fusion and dispersion by uncoupling the information provider and recipient and playing the roles including information observer, analyst and decision maker. As the Internet of Things sensor and the network based new application quantity surge, the information capacity to be processed sees an explosive growth trend. Thus, CEP technology is increasingly becoming an essential tool in many application fields. However, for most CEP engines at present, the processes and content of the complex event processing and output are open. That is to say, not only legal advanced application can utilize the CEP engine to obtain valuable information, but also illegal users are also able to acquire any necessary information for their criminal behaviors. This presents the CEP technology with huge responsibility with respect to the privacy security protection in detecting information.

Up to now, there are few studies on CEP privacy security access control, thus the research result in such aspect is just in the initial stage. In order to hold back over-class information access, literature [2] conducts security access expansion for the CEP detection and event model, which effectively prevents the unauthorized information from being leaked or tampered to the outside. It first increases two attribute fields, i.e. "security level" and "current stage", behind the traditional event model, and then adds security level checker in the query matching tree. The checker allows the event the security level of which is lower than the level set by this query matching tree to inflow so as to realize access control of the information at different security levels. Literature[3, 4]designs a set of novel security access operators and comes up with a re-query method based on such operator set with the relation algebra and query graph model of Aurora as well as the view idea of the traditional

DOI: 10.21307/ijanmc-2019-009

database management system. Through this method, the security access operators are able to be inserted to the Aurora query graph model in the most effective way. As a result, CEP can perform security access control on the data flow in pursuant to the predefined security strategy file. The above studies share the same thought, i.e. rewrite the query of the CEP event pattern, adjust the original performing structure, and insert specialized security detecting unit to form an operation structure combining security and the original detection pattern. Such kind of method is complex and has some deficiencies. 1. Users have different security strategies, thus it is necessary to save all relevant user security strategies in the security detection unit when performing multiple user security strategies in one CEP query, which obviously will cause logical mess in the course of performance. 2. The newly added security detection unit will produce more work load in the process of CEP detection and influences its execution efficiency, meanwhile the mixed operation structure will be hard to be optimized (e.g. share intermediate result). In order to avoid the above problems, this article will put forward an efficient CEP privacy security access control framework that is feasible and easy to be integrated.

II. CEP Privacy Security Access Control Object

The basic unit of CEP processing work is event. Thus, the content of its privacy security access control is the information included in the event. According to the event model definition provided by the author in the early stage of the study (Event_Model:= Event_Type @ (Attribute_Name[Data_Type]n) n≥1;), event is a tuple composed of N attributes (A1,…,An) and attribute field is the minimum unit saved by the information value. Therefore, this paper determines event attribute as the object of CEP privacy security access control and explain its concept in the form of definition.

Definition 1 Event Attribute It specifies that each event flow Stri input into the CEP engine contains one type and can only contain one type of event ETj. Certain event type ETj is made of N attributes Pk k≥1. P(Stri) represents the set of all attributes included in certain event flow Stri and P(ETj) represents the set of all attributes included in certain event type ETj, then P(Stri)=P(ETj) if ETj∈Stri. In addition, in this paper, Stri.pk (or ETj.pk) represents certain attribute in certain event flow (or certain event type).

The user's access right to the event attribute (ETj.pk|Stri.pk) content meets four cases: Completely read, partially read, access denied and quantity statistics. Therefore, a formalized description of such four types of access control operator is firstly given.

Completely read operator ξ: ξ(P(ETi))|ξ(P(Stri)) represents complete access control right to the event attribute information in the event type (or event flow). It can be abbreviated as ξ(ETi)|ξ(Stri). ξ can be used for some attributes set of the event type (or event flow), ξ(ETi[p1,p2,…]) p1,p2,…∈P(ETi) means only the information content of some attributes (p1,p2,…) in the event type is allowed to be accessed.

Partially read operator ф: ф(Expr)(ETi)|ф(Expr)(Stri) means the event attribute information in the event type (or event flow) can be accessed as per the definition of the conditional expression set Expr. The expression expri in Expr expression set only exists as conjunction relationship, e.g. ETi.location="L1"∧ ETi. temperature>30, means the location attribute of such event is L1, and the temperature value attribute is greater than 30.

Access denied operator ψ: ψ(P(ETi))|ψ(P(Stri)) represents complete denial of the access to the event attribute information in the event type (or event flow). It can be abbreviated as ψ(ETi)|ψ(Stri). Likewise, operator ψ can also only deny the access to some attributes, ψ(ETi[p1,p2,…]) p1,p2,…∈P(ETi) means only the information content of some attributes (p1,p2,…) in the event type is denied to be accessed.

Quantity statistics operator Ω: This access operator corresponds to aggregate operations that do not care the specific value of the event attribute but concern the total number, mean value and other statistics information of the event. Ω(F(Pk))(ETi)|Ω(F(Pk))(Stri) means it has statistical right to the event attribute Pk in the event type (or event flow), of which, F is calculation function, including min, max, count, avg and sum, etc.

In pursuant to the above formalized description of the control operators of security access to event attribute (record all operators set Э), the content to which the user may have privacy security access for the CEP input event flow should substantially be the result of Э operation on the input event flow by such user. That is to say, only the information in line with the given user security strategy is filtrated. Based on this, this article defines CEP privacy security access control object as follows.

Definition 2 Privacy Security Access Control Object The privacy security access control object in CEP engine is, of which, Strs is the input event flow set of the CEP engine, Э is the set of the security access control operators of event attribute, and Pi is the event attribute set in the event flow.

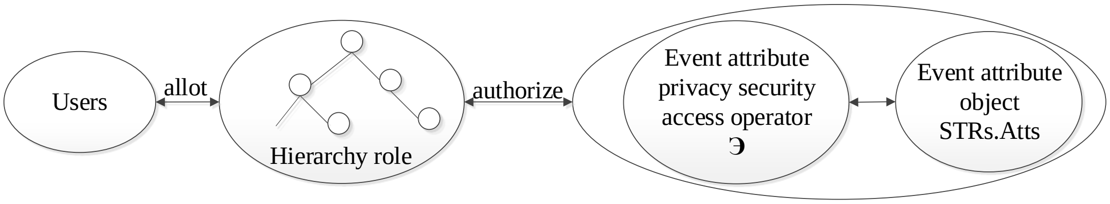

Users
allot
Hierarchy role
authorize
Event attribute privacy security access operator Э
Event attribute object STRs.Atts

Figure 1. Privacy security access control model

As shown from the above definition, when allocating security access control right to a user, the system administrator needs to explicitly designate the privacy security protection object to which such user can access for such user, i.e. designate the access right to each event attribute content for such user. This will bring huge work load for the system administrator. In order to operate flexibly and conveniently as well as reduce the work load on right allocation, this paper divides the security access control right of users based on the RBAC model and hierarchy role thought. As shown in Fig.1, hierarchy role applies tree structure. High level role may include several predecessor roles and will automatically inherit the security access rights of all predecessor roles to event attribute. Similarly, a user instance may have one or more role identities so as to realize flexible role allocation.

III. CEP Privacy Security Access Control Frame

According to the above privacy security access control object, this paper presents CEP security access control framework (CEP-SACF) as shown in Fig.2. CEP-SACF is easy to be realized without changing the original CEP implementation structure.

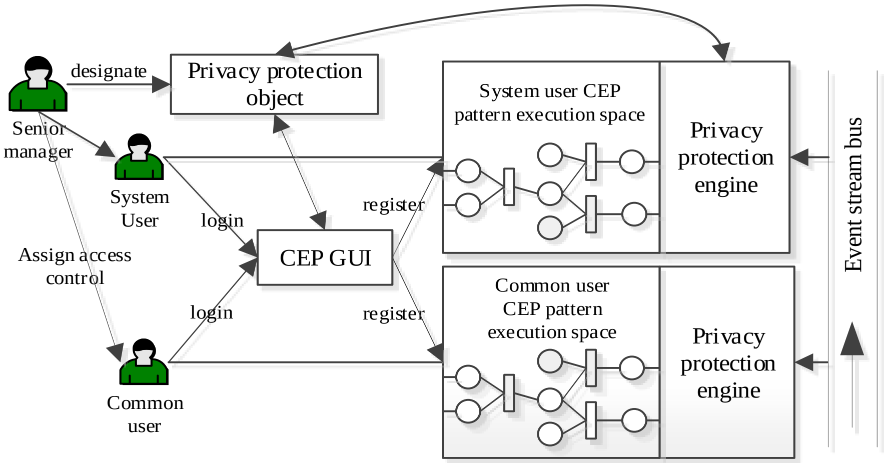

designate
Privacy protection object
Senior manager
System User
login
register
CEP GUI
Assign access control
login
register
Common user
System user CEP pattern execution space
Privacy protection engine
Common user CEP pattern execution space
Privacy protection engine
Event stream bus

Figure 2. CEP privacy security access control framework

CEP-SACF operation includes three stages, i.e. user role authorization, pattern registration and privacy security access control. First of all, at the user role authorization stage, senior system administrator defines the privacy security access control object, designates corresponding security access operation to the event attribute requiring privacy protection and allocates it to the designated user role (user role is planned by level,[5, 6]to reduce authorization work load); then, the user logs in with the role allocated and enters CEP engine management interface where user may define its own business rule with CEP event pattern language[7], CEP manages GUI and will correlate the security rules related to such user role in the privacy security access control object strategy file to detect the legality of the event pattern to be registered. If there is no conflict of security rules, then such event pattern will be registered in the independent implementation space of such user role (at the time of the first registration, an independent operation space should be firstly created for such user role, and the event pattern hereafter will be registered under the namespace with the same name as the registered user role). If the event attribute content requested by the event pattern to be accessed is in conflict with the privacy security rules of this role, then a prompt of limited user right will show and registration of such event pattern will be denied; finally, the independent privacy security protection engine will operate between the input event flow and CEP engine. Each PSPE just saves the privacy security rules related to such user role and will make operations of permission (completely read operator ξ), rejection (access denied operator ψ), filtration (partially read operator ф) or modification (quantity statistics operator Ω) for the event attribute in accordance with the definition of privacy security access operator of event attribute Э. The event processed by operator Ω will be repacked. For example, certain event contains (ID, TimeStamp, Location) attribute previously and such event only permits calculate the total number (perform Ω operation for its ID attribute). Other attributes are private information that is not permitted to be accessed. Then under the function of operator Ω, PSPE will allow all such events to pass with the private information contained flowing through the event removed. A new event only containing ID attribute will be generated. Then it will be sent to the corresponding operation space. Furthermore, in order to ensure security of CEP output result, PSPE will also receive the output in the operation space protected by it and send the result to the user within the user role of such space.

To ensure that under the registered event pattern, the user will not acquire the privacy security access right designated to such user role beyond the senior administrator and guarantee the efficiency of legal detection, this article verify the event pattern registered by the user with the following algorithm.

Algorithm 1 Validity Verification Algorithm of CEP Privacy Security Access Control in Event Pattern

```
Input: The event pattern declared by the user and user role;
Output: The event attribute array NProps[] without legal access right in the event pattern definition;
1. if (find Prop.aggregation(*) in Event_pattern)==true
       Props<String,String>.put(EventType,aggregation_operator_name);
2. Iterator (expression in where clause ) {
       EventType=get_EventType(in expression);
       Property=get_ Property(in expression);
Props<String,String>.put(EventType, Property);}
3. for(Map.Entry<String, String> entry:Props.entrySet()){
       Select * from Secunity_rule where user_role=login_user_role;
for( Dataset.hasNext() ){
if (Dataset[i].eventProperty==entry.getKey())
if (NoLegality(Dataset[i].accessOperator,entry.getValue())==ture)
           NProps[entry.getValue()];}}
4. System.out.println(NProps[]);
```

IV. PRIVACY SECURITY PROTECTION ENGINE (PSPE)

PSPE is independently created with CEP event pattern implementation space. Its content is determined by the privacy security access control rules defined by the senior system administrator and automatically updated depending on the adjustment of the rules. The basic mechanism of PSPE operation is copy, i.e. regard the event flow input into CEP engine as data bus and send the copy of the event in line with the privacy security protection rules on the data bus to the event pattern detection network inside the operation space. Beyond that, no operation will be made. This mechanism can effectively guarantee the event flow will flow through all privacy security protection engines and finally pass the event containing correctly authorized information to CEP processing nodes.

The working principle inside PSPE is shown in Fig.3. It will convert the filtration operation of the event attribute to the tree structure with the event type as the root node, of which, EventType is the event type that can be processed in this space. The subnode under the root node of the event type is the event type included. The event attribute node will be included in the access operation defined in the privacy security protection rules (as one event attribute can only define one type of security access operation type, the event attribute node only contains one subnode).

Here are some kinds of common detecting tree in PSPE. As shown in Fig.3 (a), suppose certain event type contains three event attributes and for certain user role, these three event attributes are all permitted to be accessed, thus the combined node will pass such event to the internal implementation space completely. It is contrary in Fig.3 (b) where the three attributes of the event are denied to be accessed, and such event will not be passed internally. Fig.3 (c) shows the general situation under privacy security access control, i.e. user role is only allowed to access to some attribute content of one event while the private part is not permitted to be viewed. As Attr2 attribute is denied to be accessed, the node of such detecting tree will only combine Attr1 and Attr3 attributes and outputs a new event which only contains these two attributes. Fig.3 (d) displays the appearance of the detecting tree which conditionally reads the event attribute, of which, the condition verification includes single value comparison (as shown in Figure 2 (e), the comparison content: Attr1= value 1 && Attr3!= value 2) and multiple value comparison (as shown in Fig.3 (d), the comparison content: value 2<Attr2< value 1). The node will only allow the event whose comparison result is true to pass through. Fig.3 (f) shows the situation of event attribute statistics and calculation. The node will permit such event attribute content to be accessed and the function of node Ω is equivalent to ξ. As known from the above common detecting tree structure, PSPE is able to effectively prevent the unauthorized information from inflowing and using. By means of repackaging the event, the separation of the authorized and unauthorized information can be guaranteed.

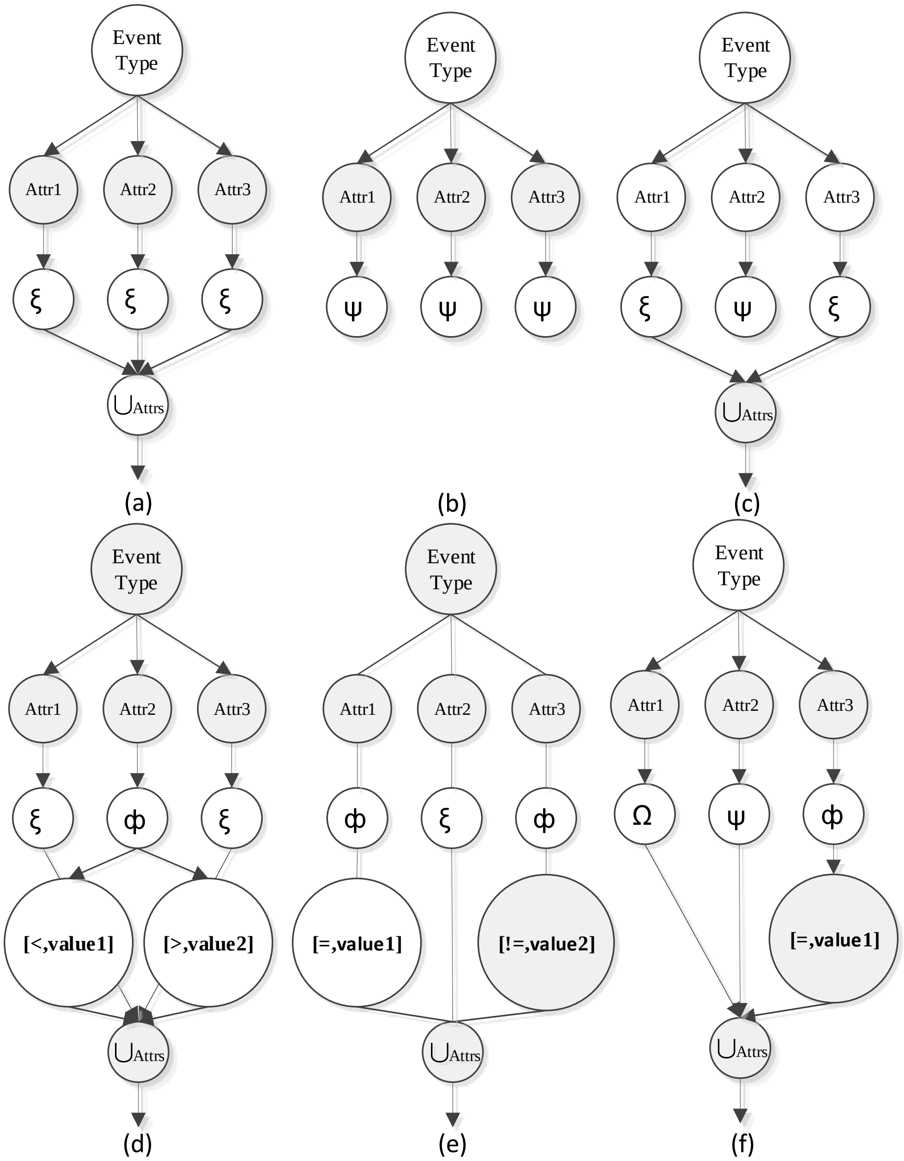
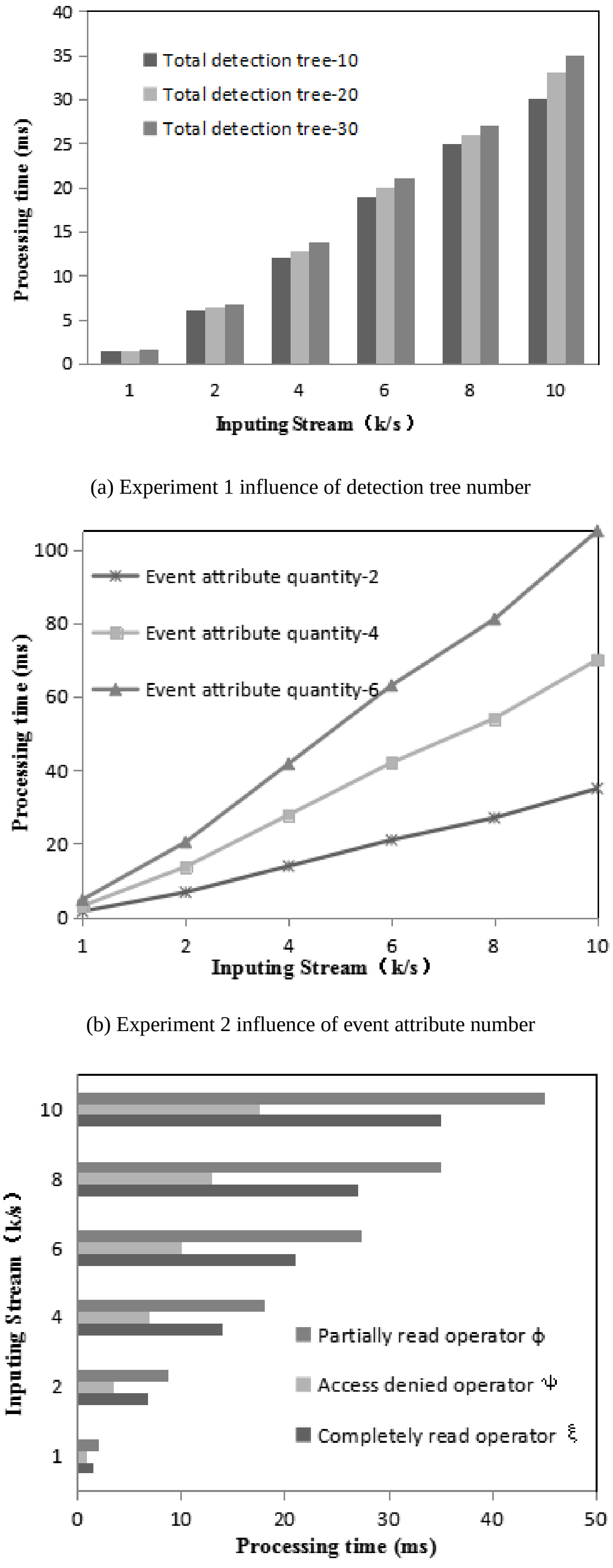


Figure 3. Detection tree structure in privacy security protection engine

V. PERFORMANCE TEST OF PRIVACY SECURITY ACCESS CONTROL FRAMEWORK

The core part of CEP-SACF operation is PSPE, the operation performance of which is related to the input event flow rate and the total quantity of the internally registered detecting tree (record such parameter as ETs). The above content shows that the working efficiency of the detecting tree is related to the number of internal event attribute node (record such parameter as ATs) and the node type of the access operator (record such parameter as OPs). Therefore, this group of experiment will test the three parameters that influence the efficiency of the engine respectively. First of all, simulate the input event flow, each of which only contains one type of event. Each event is composed of one event type attribute field and several other attribute fields. The event flow generator will utilize multiple courses to produce event flow in parallel and send it out to simulate real scene. Then, the buffer queue of PSPE will receive these events and conduct security detection by the first-in-and-first-out sequence. The detecting tree indexes with the hash table and realizes it with the custom tree structure.



(a) Experiment 1 influence of detection tree number

(b) Experiment 2 influence of event attribute number

(c) Event attribute security access operator efficiency test

Figure 4. Privacy security protection engine performance testing

Experiment I registers three groups of Ets={10,20,30} with different quantity for contrast. It receives data at 1,000-10,000 events/s and provides that the attribute quantity included in all events of such group of experiment is 2. The event attribute access operator is completely read operator ξ. As known from Fig.4 (a), as the input rate increases, the time taken by PSPE presents a linear growth trend, but the rise of total Ets has little influence on PSPE implementation efficiency because Hash Index has a high efficiency. The growth of total Ets has little influence on its index rate.

Experiment II tests the three control groups in which the attribute quantity of the event is Ats={2,4,6}. The total detecting tree registered in such group of experiment is Ets=30 (thus, the actual total number of the attribute detecting tree in PSPE is 60, 120 and 180, respectively). Also, it receives data at 1,000-10,000 events/s and provides that the event attribute access operator of all attributes is completely read operator ξ. As known from Fig.4 (b), parameter Ats has a great influence on the implementation efficiency of PSPE. As the total Ats increases, the calculated amount of the traversal node inside PSPE will undergo a cumulative rise. The processing time taken by the three control groups basically keeps a multiple relationship. The total consuming time of PSPE is at millisecond level, which has little influence on the overall operation efficiency of CEP-SACF.

Experiment III tests the performance of ξ, ϕ and ψ (as Ω and ξ is different in function, repeated test will not be done for Ω). This group of experiment provides Ets=30, Ats=2, Ops={ξ, ϕ, ψ}, with the data flow rate the same as above. The conditional expression of ϕ is [>,0]. That is to say, in spite of conditional judgment, all events are permitted to pass through. According to Fig.4 (c), as ψ denies events to pass through and there is no subsequent treatment. Thus, it consumes the shortest time (only including the time consumed in event type node searching and event attribute transversing). ϕ has calculation of conditional judgment on its node, so it consumes more time than the benchmark ξ operation. However, they come to the same conclusion that for different operators at different input rate, the total time consumed in processing by PSPE can still keep at millisecond level, which represents high operation efficiency.

VI. CONCLUSION

The experimental result shows that PSPE has a high implementation efficiency. At the same time, it is able to deal with different user roles in different ways, filtrate the event information not allowed to be accessed and generate new events in line with privacy security access control requirement. Thus, it has a feature of customizability. In addition, PSPE is completely integrated outside of CEP engine, which is very feasible because it has no influence on its original implementation structure and operation efficiency. It has certain application value by effectively making privacy security detection on the event information input/output CEP engine.

ACKNOWLEDGMENT

This work was supported by the Industrial research project of Science and Technology Department and the office of Education of Shaanxi Province(Grant No. 2016KTZDGY4-09, 17JZ004), and the Xi'an Technological University Principal Fund (Grant No. XAGDXJJ15015).

REFERENCES

[1] Luckham D. The power of events: An introduction to complex event processing in distributed enterprise systems[M]. Springer, 2008.

[2] Buddhika T, Ray I, Linderman M, Jayasumana A. Secure complex event processing in a heterogeneous and dynamic network[C]. SPIE Defense+ Security, International Society for Optics and Photonics,2014:907907-907913.

[3] Carminati B, Ferrari E, Tan K L. Enforcing access control over data streams[C]. Proceedings of the 12th ACM symposium on Access control models and technologies, ACM,2007:21-30.

[4] Carminati B, Ferrari E, Cao J, Tan K L. A framework to enforce access control over data streams[J]. ACM Transactions on Information and System Security (TISSEC), 2010,13(3):28.

[5] Tang Jin-peng, Li Ling-lin, Yang Lu-ming. User attributes oriented RBAC model[J]. Computer Engineering and Design, 2010,(10):2184-2186.

[6] Xiong Hou-ren, Chen Xing-yuan, Zhang Bin, Yang Yan. Security Principles for RBAC-based Authorization Management[J]. Computer Science, 2015,42(3):117-123.

[7] Jing Xin, Zhang Jing. Research on Parallel CEP Processing with the Multi-Event Pattern Shareing Capability[J]. Journal of Xi' an Technological University, 2014,34(9):715-719.