www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18282-18292

Reinforcement of Event Streams by Obfuscation over Multitudinous Correlations

#1 N. Pavani , #2 P. Pothuraju
Mtech Pursuing,
Assistant Professor,
Department of Computer Science and Engineering,
Vasireddy Venkatadri Institute Of Technology (Vvit) Guntur

Abstract: - Event processing is an approach of capturing and processing the data about events. The data may come from multiple origins in complex event processing systems and transmitted through multiple security authorities. Present event processing systems are failing in conserving the privacy constraints of incoming event streams in a sequence of eventually applied stream operations. This problem emerges in large-scale distributed applications like a logistic chain where event processing operators may be escalated over multitudinous security domains .This paper presents a frail access management in multi-hop event processing networks. Literally this paper offers a solution to maintain privacy constraints even when the events turn to correlated complex events. The obfuscation value calculated using Bayesian Network is used to decide whether inheritance of access requirement is needed. The implementation offers methods to enhance the obfuscation calculation and to increase the Bayesian Network size to measure obfuscation over multitudinous correlations that reinforces the event streams.

Keywords: - *Access policy, event processing, Complex Event Processing, Obfuscation Threshold, multitudinous correlations, reinforcement of streams.*

I. INTRODUCTION:

The message that intimates the change in an organization is called an event. Event processing tracks and analyzes streams of information and deduces a conclusion from them. It is important to notice failures before time in any company processes. Consider a case of manufacturing and logistics processes, where items need to be tracked continuously to notice thrashing or to redirect them during transfer. In order to satisfy this need complex event processing systems have evolved. The complex event processing (CEP) systems acknowledges to recognized meaningful events as fast as possible. CEP systems allow noticing activities by carrying out operations on event streams that come out from sensors all over the world, like packet tracking devices. In a middle way the conventional event processing systems have useful compelling operators, but the necessity to reduce the communication load in distributed network processing of stream operations have risen due to rising increase of event sources and event patrons. Additionally to this, today's joint nature of resource fallout in large scale networks allows different users, companies, or groups exchange events. As a result the mix of event processing networks in terms of processing capabilities and technologies, containing of different participants, and are expanded over several security domains. However, the security issue rose due to risen interchangeability of CEP applications. It's impossible for a middle instance to inspect access control for the entire network. Instead, each information producer must able to manage the way how data created by it should be accessed. Certain information access could be limited to a group of authorized users in a corporation. The individual event stream confidentiality and network participants authorization of security for an event-based systems already covered by the present efforts. In CEP systems there is a possibility that the event supplier may loss power on the allocation of subordinate event streams. This raises a major security difficulty that allows an opponent to figure out information of

confidential incoming event streams of the CEP systems. The opponent may be a person, group, or force that resists and or attacks.

The logistic process can be taken as an example, in which a manufacturer wants to send an item

Figure 1. Access Control & Event Dependency

To target as illustrated in Figure I.A ware house nearer to the destination is determined by the shipping company and the items will be shipped there, before delivered to the customer. The event processing system supports the logistic process. Where each party holds the operators in the field and exchange events together conceivably having private information (e.g. the shipping company receives the destination of item). If now a third party acquires events affiliated to the warehouse, it may derive conclusions about the actual event data (i.e. destination), despite of the manufacturer declaring this information as intensely secret by giving access rights only to the shipping company. The intention of this work is to build the access control that ensures the privacy of information even over numerous processing steps in a multitudinous-domain large scale CEP system.

In particular the benefits are I) an access policy inheritance mechanism to enforce access policies over a series of dependent operators and II) an ascendible method to measure the obfuscation set by operators on information that exchange between event streams. This allows describing obfuscation threshold as piece of access police, in order to specify when the event processing systems can fail in following access restrictions, therefore raising the service of the CEP system by increasing the count of events to which application components can respond to. Obfuscation means making communication disorienting that hides real meaning in communication making harder to interpret.

II. STRUCTURE MINIATURE

We imagine an interconnected dedicated host network having distributed correlations. Operators are set up on these hosts that collectively sense situations and build distributed CEP system. The directed operator graph G= (Ω, S) is used to design the supportive behavior of the operators that contains operators $\omega \in \Omega$ and event streams $(\omega_i,\omega_j) \in S \subseteq (\Omega \times \Omega)$ directed from ω_i to ω_j. Thus, for these events we name ω_i the event producer and ω_j the event consumer. One or more event attributes with different values will be there in every event. Each operator ω implements a correlation function $f\omega: I_\omega \rightarrow O_\omega$ that match incoming event streams I_ω with outgoing event streams O_ω. In particular, which events of its incoming streams are chosen, event patterns reorganization (correlated) procedure between events and the process of outgoing stream event creation is identified by f_ω. On events that are received from and originated by I_ω for the produce items in the manufacturing domain, fsc a correlation function is applied.

III. REMOTE TASK SUSTAINING POLICIES

As the usage of event-driven system increased, the efforts to make the system safe have also been increased. Consider a role based access control that planned at Pesonen et al. and Bacon et al. talk around how publish/subscribe systems protection can be achieved by inception of access control policies over multitudinous domains . The support for event communication among the domains has been explained. The perception of event owners that can be specified were presented by Opyrchal et al. and those are used to endeavor access to their events. The solution to present authentication and confidentiality for broker-less

content-based publish/subscribe system was advised by Tariq et al. This task is based on the prior work that made event communication protected between distinct entities in the system. Imagine the existence of a system that can manage access control on events. Positioned on this, in order to attain the needed access policies at any point throughout event processing, we use policy composition. Distributed systems have lot of concern in access policy composition. For composing access policies Bonatti et al. described a renowned algebra. The composition of security polices performs an indispensable role, especially in the area of web service composition, as various policies has to be associated for every grouping of web services. Our distributed CEP system yields some of these ideas that allow us to derive access limitations throughout several processing steps in the operators of our system. Techniques from statistical inference are used to understand our concepts. The Bayesian network is created and the dependencies are learned to calculate the Bayesian inference. Several Monte-Carlo algorithms have been proposed to estimate the inference value(s) as Bayesian inference is a complex calculation. From the Bayesian network probability distribution they all pick samples arbitrarily and based on samples the values are guessed. Gibbs sampler technique is followed commonly tom pick samples. The number of samples decides the accuracy of the estimated inference values.

The conditional probabilities of Bayesian network are guessed using sampling techniques. The accuracy strictly depends on the number of samples taken from the network, there is no such calculation scheme to know the polynomial time to attain certain accuracy in drawing samples. Since there is no guarantee for suitable time, the rough algorithms are made impractical for safety applications. Even though the advantage of optimizations is genuinely depends on the structure of the Bayesian network. The complication in calculating accurate inference can be concise by storing partial results of the inference estimation which otherwise would have to be calculated many number of times.

EXISTING WORK:

The proposal in existing work is role-based access control. Pesonen et al. and Bacon et al. confer how publish/subscribe systems can be secured by induction of access control policies in a multitudinous-domain architecture. They explain the way event communication between the domains can be maintained. Opyrchal et al. commence the theory of event owners that can be designated. These are used to afford access to *their* events. Tariq et al. suggest a solution that offers authentication and confidentiality for a broker-less content-based publish/subscribe system. This effort is depended on the prior work that makes event communication safe among various entities in the system. We suppose the existence of a system that can seize access control on events. Depending on this, we use policy composition in order to attain the required access policies at various points during the event processing steps.

PROPOSED WORK:

This paper addresses security in multitudinous-hop event processing networks and suggested a solution to lessen this distance. More precise, this paper suggests an approach that allows the acquiring of access requirements, when events are interconnected to complex events. The algorithm comprises the obfuscation of information, which can occur during the correlation process, and utilize the obfuscation value as a decision-making origin whether acquiring of policies is desired or not. This paper presents an accomplishment based on Bayesian Network. It enhances the obfuscation estimation methods to boost the Bayesian Network size and measures computes obfuscation over multitudinous correlations that reinforces the event streams.

IV. CEP ACCESS CONTROL

This approach allows inheritance of access requirements by passing them to event attributes in presence of an *access policy*. By this preserving of requirements through any series of dependent correlation ladder of operators in *G will be achieved*. Additional to this, an obfuscation policy allows citing an *obfuscation threshold* for event attributes. For every correlation pace the obfuscation of event attributes in created event preservatives by the planned access policy consolidation protocol. For an event attribute once the obfuscation threshold is attained, the attribute's access requirements can be ignored. In the following, we designate the concepts in the rites of access policies and obfuscation policies, and accomplish the security goal.

A) Access Policies

Access control allows designating access rights of operators (subjects) for the set of possible event attributes (objects). The owner of an object provides this access rights (e.g. the producer of an event stream) and based on an *access requirement approves to operators*. The requirement may be a role, a location or a domain group. Actually these requirements are not direct *properties* of the operators, but of the hosts at which the operators are deployed. Formally access rights are determined within an *access policy AP* of an operator ω as pairs of (attribute, access requirement):

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

Any attribute that does not hold any requirements, can be accessed by any consumer in the network. Note that attributes must be considered dissimilar even if they use the same name, but are originated at two different operators. An access requirement is an ordered pair of a property *p*, a mathematical operator *op* and a set of values *val*: *ar* = (*p*, *op*, *val*), where $op \in \{=, <, >, \leq, \geq, \in\}$. A range or a set of values specifies *val*. For the sake of clarity, access requirements are only referring to domain group in this paper and have a form like this:

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

In our sample scheme, the manufacturer's event attributes have variety of access requirements. While the item's destination information is obtainable by the customer, at the same time information about item production and picking time are confidential to the shipping company. According to this, the attached AP is defined as follows:

APmanufacturer=
{(destination,(domain,∈,*{*shippingCompany,customer*}*)),
(Picking time, (domain,=,shippingCompany)),
(place of production, (domain,=,shippingCompany))*}*

With the impulsion and assertion of access policies at each producer, a consumer can be privileged to get (receive) an attribute only if the consumer's properties and the access requirements defined for that particular attribute match with each other. In this case the consumer is trustworthy to use the attribute in its correlation function and follow the requirements specified for the attribute in its own access policy for all originated events.

B) Event Information Obfuscation

Even though access policies let a producer to signify access requirements in a graceful way, the inheritance of requirements in a series of consecutive operators is at times very confine and can restrict the efficiency and applicability of the CEP system: in each correlation step of this series, the count of access requirements may increase by the combination of requirements from multitudinous producers. Every consolidation step can then raise the count of interested consumers that are denied from access to the event attributes of originated event streams. This does not reflect the behavior of

event processing systems where fundamental events like single sensor readings might have very less impact on the outcome contained in a complex event specifying a particular situation.

In our logistics sample, *fsc* uses *destination, place of production* and *picking time* to determine the expected day of delivery. As an out come, the customer has no right to access the *expected delivery date* of the ordered item, since the customer does not match the access requirements for *place of production* and *picking time*. Yet the customer has a rational interest in this information. And one may demand that knowledge of the delivery date does not necessarily permit to demonstrate a relevant conclusion on the *place of production* and *picking time* attribute values. We can say that throughout the correlation process the attribute values get *obfuscated* and depending on the reaching level of obfuscation, the requirements to access an attribute may not needed any more. In our approach, the level of obfuscation is a degree, to which level a consumer of the originated attribute (*estimated delivery date*) can induce the value of the original attribute (place of *production).* It can be comfortably seen in the sample, that obfuscation not only depends on the attributes values, but also on the consumer knowledge. Since the *destination* value leads to the *delivery date* as well, knowledge of the destination would have great effort in trying to determine the confidential attribute place of *production* since the delivery time of the item is probably related to the destination distance and place of production. In this work, we will use *obf* (att_{old}, att_{new}, ω) to mention the obfuscation achieved by att_{new} for att_{old} when the knowledge obtainable at a consumer $\omega \in \Omega$ is given.

We allow each operator to state their access policy and also an obfuscation policy. The obfuscation policy will have obfuscation thresholds for the attributes the operator originates. Mean while in the processing of an event attribute, its obfuscation w.r.t. every possible consumer is calculated. Once, consumer reaches the obfuscation thresh hold, instead of complicated access requirements event attribute will be delivered. Officially, the obfuscation policy *OP* is named for an operator ω as a set of (attribute, obfuscation threshold) pairs:

$$OP_{\omega} = \{(att1, ot1), ..(attn, otn)\} .$$

For example, the obfuscation policy

$$O_{Pmanufacturer} = \{(destination, 0.9)\}.$$

Ignores access rights for shipping company regarding delivery date, when a specific obfuscation 0.9 is reached.

C) Security Goal

Assumption $att_{old} \rightarrow \omega\ att_{new}$ denotes the following

1) at some operator $\omega \in \Omega$, att_{old} is considered as input to the correlation function $f\omega$ and
2) $f\omega$ originates att_{new} in dependence of att_{old}. In addition, $att_{old} \rightarrow * att_{new}$ denote the

Transitive closure property of the dependency relation.

For any pair of attributes with $att_{old} \rightarrow * att_{new}$ we state that att_{new} is *dependent* on att_{old}. Our main vision is to secure the privacy of event attributes over multitudinous correlations with respect to the attributes dependency relationship originated by the CEP system. In particular, access requirements must not only be applied to the attribute $att_{old,}$ but also to be inherited to all attributes (att_{new}) that are dependent until an enough obfuscation threshold for att_{new} has been achieved. More officially, given for every attribute *att* an initialized set of access requirements are denoted by $AR_{init}(att)$. Any policy consolidation algorithm needs two conditions to be met:

Condition 1: For all attributes att $\in O_{\omega}$ originated at ω
$AR_{init}(att) \subset AP_{\omega}$.

DOI: 10.18535/ijecs/v5i10.10

Condition 2: For all pairs of dependent attributes

$(att_{old}, att_{new}) \in \rightarrow *$ with the following

1) ω_i has originated att_{old} with access requirement $AR(att_{old})$ and obfuscation threshold of $(att_{old}, x) \in op_{wi}$
2) att_{new} is originated by ω_j
3) att_{new} is used by ω_k

The access requirement in $AP_{\omega j}$ yield $AR(att_{old}) \subset AP_{\omega j}$ if obf $(att_{old}, att_{new}, \omega_k) < x$. A policy consolidation algorithm must ensure that condition 1 and condition 2 are followed in the existence of adversaries who try to acquire event attribute values they are by policy not acceptable to get directly.

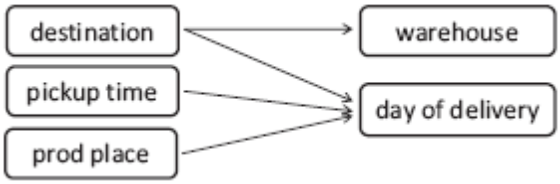

Figure Dependency Graph of the Shipping Company Operator

We would like to keep away from that hosts unrelentingly or rashly gain information from event streams for which they have no permissions. Note, by getting access to event streams according to the specific system model, hosts may still be efficient to find event attributes of illegitimate event streams from legitimate expected event streams. An adversary in our system is restricted to the behavior that described in the system miniature. The adversary is an authorized one and can access only the streams according to its properties. The final event output follows the requirements of operator and the access for every executed operator. Each adversary is clear to analyze outgoing event streams which are allowable to access, for getting any extra information.

V. EVENT OBFUSCATION FOR REINFORCEMENT

To achieve the security aim our approach set up secure event streams among every pair of operators in *G*. For setting up secure event streams we depend on mechanisms applicable in site of the art publish/subscribe systems and including our own work. In our approach it is very necessary to understand that every consumer ω_c must request essential event attributes. The requests are handled at the originator ω_p and ω_c will need to verify itself against ω_p for the correspondent event attribute. After successful verification ω_p will forwarded to ω_c

1) only those events that matches to request of ω_c,
2) only those events that contains attributes *att* s.t.
    a) the access policy of *att that* allows ω_c access to *att*,
    b) *att* has achieved a enough or high obfuscation,

i.e. $\forall(atti, ot_i) \in OP_{\omega p}\ obf\ (att_i, att, \omega_c) \geq ot_i$

To this side ω_p will have to impose on its inward streams an access policy consolidation to assure all essential access policies can be acquired and the obfuscation computation of values *obf* (att_i, att, ω_c). In the following section we will see the approach to access consolidation by miniature of all possible dependencies that are among inward and outward event streams using an event dependency graph and obfuscation policies are computed depending on a Bayesian network.

VI. OBFUSCATION CALCULATION ALGORITHM

Instead of accounting for a global Bayesian network, we suggest to exploit local knowledge available at each host. This allows us to reduce the count of relationships that are on inward.

Algorithm I Global Obfuscation Calculation

Procedure INITIALIZE(ω)

for all operator ω

do

$D_\omega \leftarrow$ FINDMULTIPATHOPERATORS(ω) end for

for all $\omega \in D_\omega$ do

$relAtts \leftarrow$FINDRELATEDATTRIBUTES for all $(att_{new}, att_{old}) \in relAtts$ do

TRANSMIT P($att_{new} | att_{old}$) TO ω end for

end for

end procedure

procedure broadcast(w)

for each w UPDATE(D_w)

end procedure

procedure PONRECEIVEEVENT(e) for all $att \in e$ do

if $\exists$ mulitPathDependency(att) then

CALCULATEGLOBALOBFUSCATION(ATT)

end if

end for

end procedure

The miniature of our advancement is that a host in the CEP network generates a Global Bayesian network for every one of its extended operators. The management (i.e. forwarding) of the event is depended on the globally attained obfuscation. This specifies that obfuscation is calculated over multitudinous correlations, and therefore certain events may be handled more reluctantly than actually preferred.

A. Measuring global Obfuscation

According to this approach, every host deals obfuscation for the globally known attribute dependencies (i.e. $att_{old} \rightarrow \omega\ att_{new}$) rather than calculating the obfuscation locally at known pair of attributes (i.e. $att_{old} \rightarrow * att_{new}$). This method has three major drawbacks: i) dependency graph will become larger ii) communication overhead may be increased, and iii) multiply connected network, though many paths of length n exists. But this method has immense benefit in comparison to drawbacks i.e. every attribute will estimate global obfuscation that the event can select one path to reach destination without estimating the obfuscation at every node. As an outcome, every host can construct a *global dependency graph* on its own as an alternative of constructing a local dependency graph for only locally dependent attributes. Moreover, we can professionally estimate the accurate implicated probability by applying erratic elimination optimization for solo connected networks to well decide the obfuscation value. Even in a local method for obfuscation estimation the multitudinous-path dependencies of attributes needed to be measured. Attributes may arrive at the recipient by means of multitudinous paths (i.e. parallel series of operators in a multiply-connected correlation networks). An adversary that can provide assurance to such attributes may be able to expect the original value by merging the event information got through different paths. For all attribute pairs with

multiple-path dependencies the operators that survive in on distinct paths swap the dependency functions w.r.t. the attributes.

B. Correctness

As our effort addresses importantly on how to initiate producer oriented access policies in CEP in an adjustable way, we furnish only familiar correctness arguments under the boundaries for the adversary. Three essential properties guarantee that the proposed method is correct in terms of definite security goal:

1) According to our expectations, an adversary tries to identify extra information by analyzing each one of event streams that can accessible. The projected algorithm includes the full knowledge the consumer *might* have. That means every attribute that influence the requested target obfuscation (obf (att_{old}, att_{new}, ω_c) accessible to the consumer must known.

2) According to Property 1, all paths from att_{old} to att_{new} are to be considered in the algorithm. That means, all parts of information an adversary may access in order to assume att_{old} is included in estimating the inference.

3) Local events that are not known (which may occur in multiple-path dependency calculations) are always handled as worst-case-deliberations. We always utilize the value in our estimations which would give an adversary the most interpreted information, i.e. the value resulting in the worst obfuscation.

While all generators of event information which might handle the obfuscation value of any operator are considered in our method, the obfuscation value proposed at an operator cannot further be lessened by any means. Hence, with the presented method, we assure that if the consumer does not obey the access requirements for an attribute att_{old}, will not be able to get any attribute att_{new} if the attributes depend on each other (($att_{old} \rightarrow * att_{new}$) unless a enough obfuscation threshold for att_{new} has been obtained. Even though, we can not assure that the consumer will receive all attributes that has attained an enough obfuscation.

VII. IMPLEMENTATION ISSUES

There are varieties of implementation issues that can be used by user to create communication. Some of the essential implementation issues are

1. ***Complex Event Processing***

CEP applies to a very extensive spectrum of confronts in information systems. Similar to business process automation and computer systems used for automated scheduling and process and processing's based on control network.

2. ***Manufacturer***

In this module the manufacturer, can insert the product details and see the request from the shipping company. Manufacturer can send details to shipping company like delivery date and picking time. These are most commonly applied to industrial production, in which raw materials are converted into finished goods on a large scale and those finished goods may be used for manufacturing other, more complex products.

3. ***Shipping Company***

In this module shipping company can see product request from customer. Then company forward these request to manufacturer or reject the request. Shipping agents will be there to take care of all the regular routine tasks of a shipping company quickly and efficiently.

4. ***Customer***

In this module, customer is the receiver of a goods, services, products, or ideas, get from a seller, vendor, or supplier for a financial or other precious consideration.

VIII. CONCLUSION

This paper addressed the acquiring and interpretation of access policies in heterogeneous CEP systems. We recognized a deficiency of security in multiple-hop event processing networks and projected a key to lock this gap. More precisely, we offered a method that allows the inheritance of access requirements, for events that are correlated to complex events. By this the reinforcement of event streams over multitudinous correlations can be achieved. Our algorithm includes the obfuscation information, which can be estimated during the correlation process, and uses this obfuscation value as a decision-making origin whether inheritance is needed or not. We presented an implementation of our method, based on Bayesian Network calculations.

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] Access Policy Consolidation for Event Processing Systems Bj¨orn Schilling∗, Boris Koldehofe∗, Kurt Rothermel∗ and Umakishore Ramachandran†∗*Institute for Parallel and Distributed Systems, Universit¨at Stuttgart.*

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] B. Koldehofe, B. Ottenwalder,¨ K. Rothermel, and U. Ra-machandran, "Moving range queries in distributed complex event processing," in *Proc. of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS)*,

[10] B. Schilling, B. Koldehofe, U. Pletat, and K. Rothermel, "Distributed heterogeneous event processing: Enhancing scal-ability 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.

[11] 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 Confer-ence on High Performance Computing and Communications (HPCC)*, 2011, pp. 355–364.

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

[13] 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.

[14] M. A. Tariq, B. Koldehofe, G. G. Koch, I. Khan, and K. Rothermel, "Meeting subscriber-defined QoS constraints in publish/subscribe systems," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 17, pp. 2140–2153, 2011.

[15] S. Rizou, F. Durr¨ and K. Rothermel, "Providing qos guaran-tees in large-scale operator networks," in *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*, 2010, pp. 337 –345.

[16] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice Hall, 2002.

[17] S. Geman and D. Geman, "Stochastic relaxation, gibbs dis-tributions, and the bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, pp. 721 –741, 1984.

[18] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.