Content-Based Access Control: Use Data Content to Assist Access Control for Large-Scale Content-Centric Databases

Wenrong Zeng, Yuhao Yang, and Bo Luo
Department of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center
The University of Kansas, Lawrence, KS 66045, USA
Email: wrzeng@ittc.ku.edu, yyang@ittc.ku.edu, bluo@ittc.ku.edu

***Abstract*—In conventional database access control models, access control policies are explicitly specified for each role against each data object. In large-scale content-centric data sharing, it might be difficult to explicitly identify accessible records for each role/user, especially when the semantic content of data is expected to play a role in access decisions. As a result, users are often over-privileged, and *ex post facto* auditing is enforced to detect misuse of the privileges. Unfortunately, it is usually difficult to reverse the damage, as (large amount of) data has been disclosed already.**

In this paper, we introduce Content-Based Access Control (CBAC), an innovative access control model for content-centric information sharing. CBAC is expected to be deployed on top of Role-Based Access Control (RBAC) or Multi-level Security (MLS), in the application scenarios where RBAC and MLS will give excessive access rights. As a complement to conventional access control models, the CBAC model makes access control decisions based on the content similarity. In CBAC, each user is allowed by an MLS or RBAC rule to access a large set of data objects, while the CBAC rule imposes an additional layer of restrictions that the user could only access "a subset" of the designated records. The boundary of the subset is dynamically determined by the textual content of data objects. We then present an enforcement mechanism for CBAC that exploits Oracle's Virtual Private Database (VPD). To further improve the performance of the proposed approach, we introduce a content-based blocking mechanism to improve the efficiency of CBAC enforcement. We also develop a content annotation mechanism for more accurate textual content matching for short text snippets. Experimental results show that CBAC makes reasonable access control decisions with a small overhead.

***Keywords*-Content-based Access Control, Database Security**

I. Introduction

Database access control models define "who can access what", where "who" represents a set of users/roles, and "what" represents a set of data objects, e.g. tuples or XML nodes. Meanwhile, access control enforcement mechanisms precisely and efficiently implement the models on real data. In conventional database access control models, administrators or data owners explicitly specify access rights of each data object for each role (e.g. using GRANT/REVOKE). However, such approaches may not be suitable for content-centric data, where data content is expected to play a role in making the access decisions. In particular, it could be difficult to *explicitly* describe access rights for very large amounts of data objects, especially when the decisions are based on content – it is too labor-intensive to require a system administrator to manually examine every record in the database and assign access rights to each user/role. As a result, access control management becomes too labor-intensive, or coarse-grained access control policies are employed so that users are over-privileged. To further motivate this research, let us see the following examples:

Example 1. A law enforcement agency (e.g., FBI) holds a database of highly sensitive case records. A supervisor assigns a case to agent Alice for investigation. Naturally, the supervisor also needs to grant Alice access to all the *related* or *similar* cases. The concept of "related cases" is determined by the semantic content of the records, which could be geological, temporal, *motis operandi*, or just the similarity in the textual description of the case records. Moreover, when new cases are added to the database, the ones that are similar to Alice's should be automatically made accessible to Alice, without requiring the supervisor to further intervene. Unfortunately, this type of access control description is not supported in the existing database access control paradigm. In practice, the MLS or RBAC models are adopted and every agent is granted access to a very large number of records. Another approach is to allow Alice to access briefs of all cases. She needs to request for authorization (from the supervisors) to access the case details, if a case appears to be "related" from the brief.

Example 2. Healthcare information sharing is strictly governed by HIPAA. Medical records are well protected by healthcare providers, and are only shared under very rigorous rules. However, within the facility, users (doctors, nurses, researchers) are often given broader access privileges, while *ex post facto* auditing is enforced to detect and punish misuse of the privileges [1]–[4]. Another thrust of solutions employs the "break the glass (BTG)" mechanism – to allow users to break access control rules in a controlled manner in special circumstances [5]. Additional auditing will be performed once a user invokes the BTG policy.

Both examples demonstrate applications where explicit access control specifications at record level are too labor-intensive; therefore, users become significantly over-privileged due to the nonexistence of record-level content-based access control. The excessive privilege is somehow mitigated with two controls: (1) RBAC or MLS is enforced so that users have basic clearance to access the database; and (2) *ex post facto* auditing is enforced to punish misuse of the privileges. However, with the size of data, the basic clearance still allows a user to access an unacceptable amount of records. Meanwhile, *ex post facto* auditing does not reverse the damage, since the suspicious user has already committed the misfeasance, and it is impractical to revoke disclosed data. Ideally, we expect a more restrictive and automated access control model, instead of allowing users to be significantly over-privileged or requiring excessive human intervention. That is, the new model is expected to intelligently identify a smaller subset of records that are relevant to the user's task, and only grant access to this subset.

Attribute-based access control (ABAC) could be employed to partially mitigate the problem. For instance, in Example 2, we can specify access control based on a combination of doctors' and patients' attributes: a doctor may access records of patients that have ever been treated in his/her department. However, attribute-based access control may not work with unstructured text (free text) content. Moreover, when the database structure and the attributes are very complicated, it may be difficult to obtain closed-form expressions for ABAC policies. In such applications, "hard security" requires a high price of excessive human labor and degradation of usability (e.g. waiting for manual authorization in BTG). From the technology perspective, there does not exist a computational model to precisely describe semantic content, or to model this human cognitive process – the rationale behind the decision is too vague and complicated.

In such use cases, it is expected to have an access control model that extends ABAC to make access decisions based on the *semantic content* of the data. It is also desired that such content-based access control capability to be provided by RDBMS as native functions, and only requires minimal intervention from administrators. In this paper, we present our first attempt towards this endeavor: we introduce the *content-based access control* model and enforcement mechanisms. In particular, we propose a two-phase hybrid solution: (1) the data owner or administrator manually identifies a small *base set* of records – the core of the set of records that are accessible to the user; and (2) at runtime, CBAC extends the base set and makes access verdicts according to specified CBAC rules, which are based on the *lexicon similarity* between the base set and the requested records. The new model, as an extension to ABAC and a complement to legacy access control approaches, provides an effective and efficient means of access control that exploits content features in content-rich data sharing.

We would like to emphasize that **content-based access control does not imply weakened or relaxed security**. Rather, it enforces an additional layer of access control on top of existing "precise" access control methods. CBAC allows approximation – it does not provide a static boundary for the accessible set of records. However, allowing the user to access a small set (size of n) of roughly (and automatically) selected records is **more secure** than allowing the user to access all the records in the pool (size of N), especially considering that n is usually orders of magnitude smaller than N.

Our contributions are three-fold: first, we formally propose a data-driven access control model that exploits the data content to achieve flexible and powerful access control semantics. Second, we develop an effective enforcement mechanism of CBAC utilizing native functions from off-the-shelf database systems. Last but not least, we further develop a blocking mechanism and a labeling mechanism to improve the efficiency for CBAC enforcement, and to improve the accuracy of textual content matching.

II. The Problem

To satisfy the needs for describing and enforcing access control without explicitly identifying every subject and object, we propose *content-based access control* (CBAC), as an extension to ABAC and an addition to the legacy database access control models. CBAC works best for content-rich data sharing scenarios with the following assumptions:

(1) Basic privileges: Users must be authenticated with basic trust, for instance, through RBAC or Multilevel Access Control. As in Example 1, Agent Alice must have "classified" authorization in order to access all classified records. CBAC further restricts users' access rights to a smaller set of records.

(2). Data-driven access decisions: There are large amounts of data objects, and the data is content-rich in nature. Each data object features a block of unstructured textual content (e.g. a CLOB type attribute). The access control decision for each user against each data object is expected to be data-driven (content-driven). In particular, the decision is supposed to be determined by the content of the textual data, as we have illustrated in previous examples.

(3). Lack of explicit authorization: explicit authorizations (for all users against all data objects) are not available, as it requires excessive labor to examine the textual content for each record and make a verdict for each user.

(4). Approximation: approximation is allowed in the application – it is acceptable if a user (of the special role) accesses a few more (or a few less) records than it would have been assigned by an administrator. Note that such approximation does not implicate relaxed security.

Example 3. If we revisit Example 1: assume that only 15 cases in the database are relevant to Alice's case, that says, a careful and accurate supervisor would only allow Alice to access those 15 records. However, if an automated mechanism blocks a small portion of these 15 records, or allows Alice to access a few other records, it is considered to be acceptable, especially comparing with the current practice which gives Alice access to all the records.

Our goal is to design an access control model that considers the textual content of data objects in making access control decisions, and to develop a mechanism that enforces this model efficiently. Meanwhile, the access control enforcement mechanism is expected to have the following features:

(1) Autonomous: CBAC enforcement mechanism is expected to require minimal intervention from the system administrators and data owners.

(2) Transparent: users are expected to issue queries as usual – without being affected by the existence of the access control mechanism.

(3) Efficient: although content similarity assessment could be computationally expensive, we still expect the CBAC enforcement mechanism to return answers promptly.

(4) Off-the-shelf: the CBAC enforcement mechanism is expected to employ native access control capabilities from off-the-shelf database management systems, so that the proposed model and mechanisms could be easily adopted.

III. The Content-Based Access Control Model

A. Initial authorization and the base set

A simple access control policy could be specified as a 4-tuple:

$$ACR = [subject, object, action, sign] \tag{1}$$

where the *subject* denotes the user, the *data object* could be a table, an attribute, or a tuple in the relational data model. The *action* identifies an *operation* on the data object, such as read or update, and the *sign* denotes if the operation is allowed or denied. We assume *fine-grained access control*, in which access control is enforced at record or node level. Hence, we consider each tuple in the relational model as a data object. Hereafter, we assume relational data, and we use terms "record" and "data object" interchangeably. In conventional access control models, a definitive authorization is required for each user against each object. The authorization could be role-based or attribute-based. Records without an explicit authorization is considered to be "access denied". Therefore, the data owners or administrators need to identify all accessible records for each user, and define explicit rules for such records.

In CBAC, we propose a two-phase access control specification model for content-centric information sharing. First, in the *initial authorization* phase, a user is given access to a small set of records, denoted as the *base set* or the *seed records*. They are explicitly declared in CBAC, to be different from records that are authorized through regular authorization. The base set could be selected in different ways: (1) they could be manually selected by the data owner or administrator, e.g., the supervisor assigns a case to Alice in Example 1. (2) Seed records could be identified with attribute-based rules, e.g., a doctor's base set includes all his/her own patients. This method is efficient, but may not be available in all applications. (3) Alternatively, we can allow users to request for base set items, and ask the administrator to approve the request.

In addition, in initial authorization, the administrator also identifies a set of records that are subject to CBAC authorization – namely the *CBAC candidate set*. CBAC cannot be used to grant access to records outside of this set. This requirement could be specified with conventional RBAC or ABAC.

B. Content-based authorization

In the *content-based authorization* phase, the CBAC mechanism automatically expands the base set according to pre-specified CBAC rules, in which authorizations are defined upon content similarity between requested records and base set records. That is, the static, binary notion of $sign \in \{\texttt{true}, \texttt{false}\}$ in ACR policy (1) is extended to be a content-based access control function, which is evaluated during access control enforcement. A *generic CBAC policy* could be:

$$ACR = [subject, object, action, f(\mathbf{s}, \mathbf{d}_i)] \tag{2}$$

where the *object* denotes the CBAC candidate set, s represents the *base set* for the subject, and $\mathbf{d}_i$ represents a non-seed data object that needs authorization from CBAC. For a user query, the function evaluates to a value $f(\cdot,\cdot) \in \{\texttt{true}, \texttt{false}\}$ for each $\mathbf{d}_i$. Access to the record is granted when $func(\cdot,\cdot) = \texttt{true}$. The decision function should consider two main factors: *content modeling*, and *content-based similarity assessment*. A generic decision function $f(\cdot,\cdot)$ will get the maximum similarity between the data object against all seed records of the subject, and compare with a preset threshold:

$$f(\mathbf{s}, \mathbf{d}_i) = (\max_j(SIM_d(\mathbf{s}_j, \mathbf{d}_i)) \geq T) \tag{3}$$

In general, the similarity between two records is defined as the weighted sum of the similarities across all M attributes:

$$SIM_d(\mathbf{d}_i, \mathbf{d}_j) = \sum_{x=1}^{M} \omega_x \times sim_{a_x}(d_{i,x}, d_{j,x}) \tag{4}$$

where sim_{a_x} is the normalized similarity function defined on the domain of a_x, while ω_x is the weight on the attribute. When we only consider simple types such as integer, the similarity functions could be relatively trivial, e.g., subtraction of two values. However, in CBAC, we are more interested in content-rich unstructured text data. In this paper, we consider the baseline model for content-rich types: (1) data objects are represented by vector-space or annotation-based models; and (2) the content-based access control decision is made by evaluating the *lexicon similarity* between seed records and data objects. Note that CBAC creates a many-to-many relationship between users and records, i.e., a record is accessible to many users. This is considered to be suitable for many content-rich information sharing applications, such as the ones we used in the examples.

Example 4. Let us assume that content-based access control is adopted in Example 1. In the initial authorization phase, the supervisor explicitly assigns cases to agent Alice, to add them to Alice's base set. Alternatively, agent Alice may submit a new case, or request to access a case by reviewing public attributes of the case, and the supervisor will have to approve the submitted or requested cases before they are added to Alice's base set. Meanwhile, the supervisor specifies the following CBAC policy for all agents at `classified` level:

$$ACR = [sbj, \mathbf{d}_i \in \mathbf{D}_{\text{cls}}, \text{read}, \max_{\mathbf{s}_j \in \mathbf{S}}(SIM_d(\mathbf{s}_j, \mathbf{d}_i)) \geq T] \quad (5)$$

That says, the CBAC candidate set includes all the cases at `classified` level ($\mathbf{D}_{\text{cls}}$). User sbj is granted "read" access to a record $\mathbf{d}_i$, when the content similarity between $\mathbf{d}_i$ and any one of sbj's records $\mathbf{s}_j$ is greater than a preset threshold. This rule will be enforced in the content-based authorization phase.

C. The content model

We have defined an abstract similarity function $sim_{a_x}(d_{i,x}, d_{j,x})$ for attribute a_x. In practice, implementation of $sim_{a_x}()$ is expected to be adapted to different application scenarios and data types. As we have discussed, the CBAC model is mostly designed for content-rich unstructured text types, such as VarChar and CLOB. Ideally, we are expected to model such types by their *linguistic semantics* (linguistic content), however, natural language understanding remains an open problem, and it is very difficult to provide a reliable similarity assessment purely based on the linguistic content [6], [7]. While the specific attribute similarity measurement is not the focus of CBAC, in this paper, we start with the classic TF-IDF model, which is generic measure that works for unstructured text documents. We further present a annotation-based model to tackle the lexical ambiguity issue, and improve the accuracy of content matching.

We model unstructured text content by the statistical distribution of terms. The terms (words) from the selected textual attribute for all tuples are collected to construct a feature space (term space). For each record, the content-rich attribute is represented as a vector in the term space: $\mathbf{d}_i = [w_{1,i}, w_{2,i}, ..., w_{N,i}]$, where $w_{t,i}$ is the TF-IDF weight of record i on term t. The original TF-IDF weight is defined as: $w_{t,i} = tf_{t,i} \times idf_t = tf_{t,i} \times \log \frac{N}{df_t}$, where $tf_{t,i}$ is the frequency of term t in record i, and df_t is the number of records that contain term t. Many variations of TF-IDF have been used in the research community [8]. Furthermore, the similarity score between two records is calculated as the cosine similarity of two record vectors: $sim(\mathbf{d}_i, \mathbf{d}_j) = \frac{\mathbf{d}_i \cdot \mathbf{d}_j}{|\mathbf{d}_i| \times |\mathbf{d}_j|}$.

The threshold in the CBAC decision function (3) is selected based on knowledge of the text corpus, the similarity function, and the access control expectations of the application. In conventional TF-IDF model with cosine similarity ($sim \in [0, 1]$), a smaller threshold will allow the user to access more documents.

Please note that the choice of content modeling and similarity measurement is not determined by the CBAC model, rather, it is the choice of the system administrators or data owners, who specify the policies. In Section IV, we enforce CBAC with the content model embedded in Oracle. Moreover, in Section 5, we employ a content-based tagging approach to tackle the sparseness and polysemy (lexical ambiguity) issues with short text snippets in the TF-IDF model.

D. Top-K similarity

In the basic CBAC model, content similarity is compared with a preset threshold, and the user is granted access to all of the "similar records". A potential problem is that the number of accessible records depends on the preset threshold. When the administrator is unfamiliar with the data, he/she may assign a bad threshold, which gives the subject access to too many or too few records. To tackle the problem, *top-K similarity* could be used. Instead of setting a threshold for record similarity scores, the administrator could preset the number of data objects to grant access. For instance, in Example 1, we may define that Agent Alice is allowed to access 300 cases that are most similar to her seed cases. The top-K similarity measure provides flexible and intuitive control to database administrators and data owners.

E. Security and usability analysis

Privilege escalation. When CBAC is not properly enforced, privilege escalation might be a threat. That is, a user may first get access to a set of records ($\mathcal{C}_1$) that are similar to the seed records, and then attempt to claim $\mathcal{C}_1$ as seeds to gain access to additional records ($\mathcal{C}_2$, where $\mathcal{C}_2 \supset \mathcal{C}_1$). This type of privilege escalation is not allowed in the CBAC model – only privileged users (data owners or administrators) are allowed to add items into one's base set, while records accessible through CBAC ($\mathcal{C}_1$) cannot be claimed as seeds by the user. Moreover, in DAC settings, when data owner U_1 adds his records to U_2's base set, such records are only expandable within data owned by U_1. That is, U_2 cannot use U_1's records as seeds to access similar content owned by other users.

Content forgery attacks. Adversaries may purposely manipulate his/her base set, by updating existing records or uploading new records – the *content forgery attacks*. In CBAC enforcement, base set items are only assigned or approved by privileged users. Meanwhile, updates to base set records are either disallowed, or need to be re-approved by the administrator before they take effect in CBAC enforcement. Hence, content forgery attacks are not effective in CBAC.

Security guarantee. The security of CBAC relies on the security of the base set. In summary, although CBAC is primarily designed for low-security applications that allow approximation, we still provide the following security guarantee: *when CBAC is correctly enforced and managed, a malicious user cannot obtain access to sensitive information by manipulating his/her accessible records, creating spoofing records, or gaining (non-base-set) access to similar insensitive information.*

Usability analysis. In CBAC, the data owners or administrators only need to explicitly specify or approve the *base set* records for each user. The required effort is significantly less than manually authorizing every record against every user, since the administrators only manage the "core" records (base set) for each user. CBAC automates access control specification for all other records, and hence eliminates most of the effort from the administrators. Moreover, when new records are added to the database, CBAC instantly grants access to qualified users, without any intervention of the administrator. The administrator only needs to manually intervene if the new records need to be added to someone's base set. In summary, CBAC improves usability by significantly reducing the workload of manual access control specification, and making new records instantly available to qualified users.

Example 5. Content-based Access Control could be easily adopted in the scenario described in Example 2 to replace the break-the-glass mechanism. In CBAC, the original access control rules are defined as the *initial authorization*, while the function of the BTG mechanism is served by the content-based authorization.

IV. CBAC ENFORCEMENT

A. CBAC with on-the-fly similarity assessment

In CBAC enforcement, we exploit Oracle's VPD modula to implement record-level access control. To enforce CBAC in VPD, we rewrite the user's query by appending a dynamic predicate, which represents the content-based access control semantics. The range of accessible records is determined by the user's base set, as well as the similarity-based access control function. As we have introduced, there are two types of CBAC policies: (1) threshold-based CBAC, and (2) Top-K CBAC. In this subsection, we assume that similarity assessments are performed on-the-fly.

Settings. In the experiments, we utilize the NSF research awards data set [9]. Awards are extracted, parsed and loaded into three tables: Award_Basic (A_ID, Title, A_Instr, Div, abs, S_date, E_date, Ex_tol_amt); Aw_Intr(A_ID, I_ID); Investigator(I_ID, I_Name, I_Email). In particular, attribute `AWARD_BASIC.abs` contains full-text abstracts of NSF awards, representing the content-rich information. To demonstrate the scalability of CBAC enforcement, we increase the number of records in the database by adding synthetic dummy records. We employ an automatic CS paper generator *SCIgen*[1] to generate very large amount of content-rich but meaningless records. Eventually, the database has 2,714,025 records. Note that the content in this database is not sensitive thus does not require access control, however, it mimics content-centric databases, for which CBAC is designed.

We use Oracle 11g for the experiments with CONTEXT indexing. The experiment runs on a 64-bit Windows 7 system, with Intel® Core™ 2 Duo CPU E8500 @ 3.16GHz and 4.0GB RAM. Queries are issued from SQL-Plus, and the evaluation time includes all I/O (e.g. network I/O).

Experiments. We mimic the scenario in Example 1. In initial authorization, the *base set* of each user is defined as the award records PI-ed by the user. Each user is explicitly granted access to such records. Next, we simulate the following access control scenarios: (R1) an attribute-based access control (ABAC) rule: the user is allowed to access records in a division where he/she has PI-ed an award; (R2) a content-based access control (CBAC) rule: the user is only allowed to access awards that have similar abstracts with the awards in his/her base set; and (R3) a combined (ABAC+CBAC) rule: R1 AND R2. All three scenarios are implemented with Oracle VPD.

In the experiments, we login as 60 randomly selected users with the following queries.

```
QUERY1: SELECT TITLE, ABS FROM
        johndoe.AWARD_BASIC
        WHERE S_DATE >=
        TO_DATE(’1996/01/01’, ’yyyy/mm/dd’)
        AND   ROWNUM<=10;
QUERY2: SELECT COUNT(*) FROM
        johndoe.AWARD_BASIC
        WHERE S_DATE >=
        TO_DATE(’1996/01/01’, ’yyyy/mm/dd’);
```

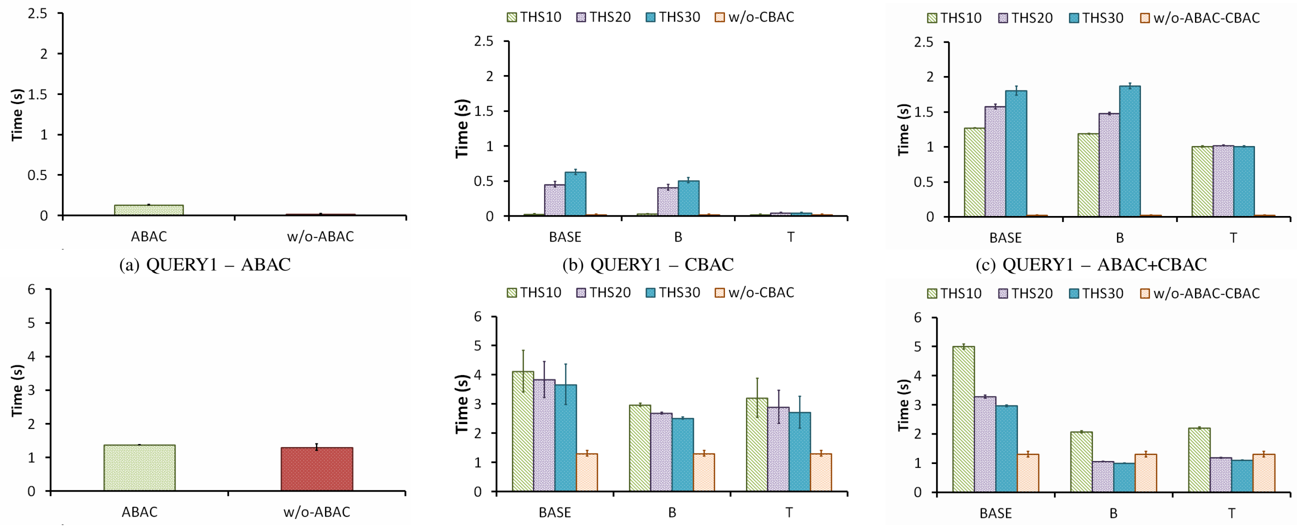
The end-to-end evaluation time for ABAC (R1) is shown in Figure 1 (a) and (d). For threshold-based CBAC, the evaluation time is shown in Figure 1 (b), (c), (e) and (f) "`BASE`". Note that the rightmost bar in this group indicates the "no-CBAC" case, where no access control is enforced. We have performed the experiment with different thresholds – a larger threshold means a stricter constraint, which requires higher similarity between the queried records and the seeds. As shown, query processing for Query1 with CBAC is very efficient. A larger threshold leads to slower query processing, since Oracle needs to scan through more records to identify first 10 records that satisfy the stricter CBAC condition. Query evaluation slows down with R3, with the overhead required by both ABAC and CBAC semantics. On the other hand, Query2 forces Oracle to go through all records. As shown in Figure 1 (e) and (f), the overhead is acceptable, especially consider that CBAC models data content in a high-dimensional vector space, which requires excessive computation.

Top-K CBAC. We have developed two implementations of top-K CBAC.

Naive implementation. In an naive implementation, we simply included the top-K semantics in the dynamic predicate. Unfortunately, query performance was very slow, since the top-k ranking in VPD predicate was repeatedly evaluated.

Optimized implementation. To improve query performance, we split the top-K semantics into two steps: (1) in PL/SQL, we select the top K records that are most similar to the base set; (2) we identify the similarity score (T_s) of the K-th record, and generate a threshold-based predicate with threshold T_s. The average end-to-end processing time (Figure 2 (a) '`BASE`') is significantly reduced. Note that query evaluation is still relatively slow comparing with threshold-based CBAC, mainly due to the size of the database. Especially, Oracle does not provide native support for selecting first K records – Oracle sorts the entire table to return the top K records (complexity: $O(N \log N)$). However, the computation of selecting and ranking top K records could be as low as $O(N + K \log K)$ [10]. In Section V, we will optimize top-K CBAC performance

[1] Available at: http://pdos.csail.mit.edu/scigen/



(a) QUERY1 – ABAC

(b) QUERY1 – CBAC

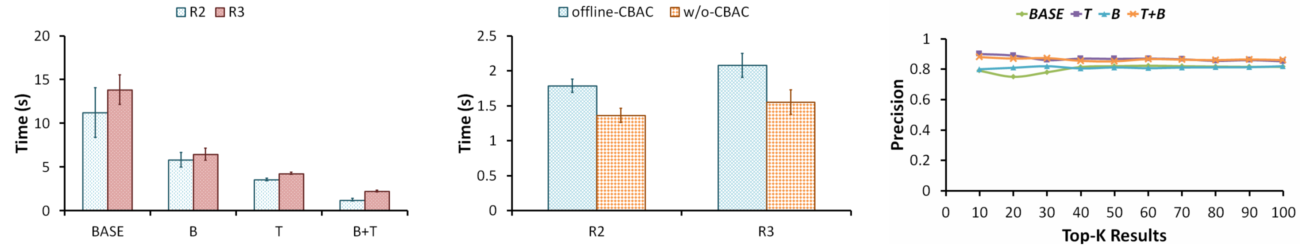
(c) QUERY1 – ABAC+CBAC

(d) QUERY2 – ABAC

(e) QUERY2 – CBAC

(f) QUERY2 – ABAC+CBAC

Fig. 1: End-to-end query processing time for threshold-based CBAC. "BASE": baseline CBAC; "T": CBAC with tagging; "B": CBAC with cloking; and "T+B": CBAC with tagging and blocking.



(a)

(b)

(c)

Fig. 2: (a) Top-K CBAC query performance; (b) offline CBAC query performance; (c)Soundness of CBAC Enforcement.

using blocking and tagging.

B. CBAC with offline similarity assessment

In some applications, the frequency of data update is much lower than the frequency of data access, where the database is only updated when new cases are added or updated. In such applications, offline similarity assessment could be employed to improve the efficiency of CBAC enforcement. In this approach, the accessible records for each user are identified offline given a specified threshold or a threshold value corresponding to a certain K.

We use Query2 to evaluate CBAC with offline similarity assessment, and compare with the "no-CBAC" case (in "R3 with no-CBAC" case, ABAC is still enforced). As shown in Figure 2 (b), queries with offline similarity assessment only introduce a 20% overhead.

V. Content-based blocking and labeling

In this section, we discuss two optimization approaches for CBAC enforcement. First, we aim to improve the efficiency with content-based blocking. Next, we improve the accuracy of content similarity assessment.

A. Content-based blocking

To improve CBAC performance, we propose the content-based blocking scheme. When the similarity function $SIM(\cdot,\cdot)$ is provided with the access control policies, we pre-partition the records into c non-overlapping clusters, so that records with similar contents are labeled in the same cluster. The centroid of cluster C_k is defined as the mean document of the class: $\mu(C_k) = \frac{1}{|C_k|}\sum_{d_i \in C_k} \mathbf{d}_i$. The centroids of all clusters are stored in a separate table. After clusters are created, they could last for a significant period of time – when records are inserted or updated, we only update the related clusters, which is very efficient.

First each incoming query is evaluated against the cluster centroids, to identify the most similar x clusters, where $x << c$. Second, the query is only evaluated against the records in the selected x clusters. That is, we add a new predicate, which requires the records to have one of the x labels. With blocking, we expect to improve query performance without sacrificing usability.

Experiments. We employ the enhanced *K-means* clustering tool from Oracle Data Mining to pre-process the records.

We set the number of clusters: $c = \sqrt{\frac{n}{2}}$, where n is the number of samples. Clustering 2M records and storing takes approximately 3 minutes in our experiment. Query performance for threshold-based CBAC with blocking is shown in "B" bars in Figure1. Performance improvement appears to be limited, since the CONTEXT indexing was very effective in the basic approach (the BASE bars). Meanwhile, the query performance for top-K CBAC (shown in "B" bar in Figure 2 (a)) is improved, since we only sort a smaller number of records.

B. Content-based labeling

As introduced in Section III, the CBAC could take any attribute-similarity measurement. While the vector space model is the most popular method in information retrieval applications, it suffers from drawbacks of the bag-of-words model. As an example, let us look at the following short documents:

```
D1: privacy preserving similarity
    assessment for semi-structured data
D2: private XML document matching
```

It is clear that D1 and D2 are both about the same topic. However, in vector space model, D1 and D2 are orthogonal. To tackle the problem of lexical ambiguity, we employ annotation-based approaches to represent documents in an unambiguous "topic space". In this paper, we utilized TAGME [11] to form "bag-of-topics" model. The overhead for tagging is insignificant, consider that it is performed only once for each record. Tagging for large amount of existing data is performed offline, while online tagging is only employed for new or updated records.

Experiments. In the experiments, we annotate every abstract with related *topics*. Each topic is associated with a "confidence factor" ρ in the range of 0 to 1, which reflects the quality of the annotation. To maintain the quality of tagging, we fit all the ρ to a non-parametric distribution, and observed that by setting the threshold into 0.2, 80% of the tags are removed (Pareto principle, a.k.a. 80-20 rule). The filtered topics are added to a new CLOB attribute (with CONTEXT indexing) in the table AWARD_BASIC. In the new topic space, noises in term distributed data has been removed.

We first evaluate threshold-based CBAC with tagging, as shown in "T" bars in Figure 1. Query processing performance is improved with the shorter length of "bag-of-topics" model. Meanwhile, for top-K CBAC, the experimental results are shown in "T" bars in Figure 2 (a). The performance is also improved, since a very large portion of records have "0" similarity with the seeds, and they are eliminated in sorting. For top-k CBAC, we attempt to combine blocking and tagging. As shown in "T+B" bars in Figure 2 (a), the query performance is again improved, and the end-to-end query evaluation time for top-K CBAC now becomes very acceptable – only slightly slower than the "no-CBAC" case in Figure 2 (b). The results confirm that CBAC is fast enough to be adopted in real-world applications.

C. Soundness of CBAC Enforcement

CBAC is said to be *sound*, when a CBAC enforcement mechanism makes access control decisions that are consistent with users' decisions. However, it is impractical to evaluate the relevance of a record against 100K records. Therefore, we attempt to evaluate the top-100 records identified by CBAC to assess if a DBA would agree with CBAC's access decision. In the experiments, we first use three rules to coarsely identify "relevant records". We noticed that every record in NSF database is assigned with a set of *field identification numbers*. If two records share one or more field identification number(s), they are initially considered to be relevant. Besides, if the two records share two closely related field identification numbers, they are considered to be relevant. Last, if the seeds' content show a close relationship to the record's field name, they are considered to be relevant. Finally, we manually examine all the "relevant documents" identified by these rules, and eliminate the ones that appear to be irrelevant to us. We have tested queries from 60 different users in different disciplines including biology, chemistry, mechanical engineering, mathematics etc, and measures the precision of top-K results for all the queries. As shown in Figure 2 (c), the user would agree with approximately 80% of CBAC's (positive) decisions. Again, the tagging approach improves CBAC accuracy.

Please note that the accuracy of the content similarity measurement is not a research problem in the security community. Rather, we are utilizing the methods from information retrieval and NLP communities. Any content modeling and similarity assessment method could be used in CBAC.

VI. Related Work

Database Access Control. Database access control research could be roughly categorized as *access control models* and *access control enforcement*. Here we provide a very brief introduction. Relational access control models can be classified into: *mandatory access control* [12], [13], *discretionary access control (DAC)* [14], [15] and *role-based access control (RBAC)* [16]. Most real world RDBMS implement a table/column level DAC or RBAC similar to the one in System R [17]. View-based approaches are traditional methods to enable row-level access control [18], [19]. Over the years, many models and enforcement mechanisms have been proposed (a survey is available at: [20]), such as the Flexible Authorization Manager (FAM) [21], temporal DAC and RBAC models [22], [23], credential-based access control [24], [25], group-centric models [26]; and more recently: purpose based access control [27], policy-based access control for the semantic web [28], [29], and access control for the Web [30], [31]. Much effort has been devoted to facilitate effective management of users, roles, rules in different applications, e.g. rule-relationship analysis [32], role mining and administration [33], [34], policy integration and user provisioning [35], [36], mediation in distributed systems [37].

Content-based Access Control. In [38], Bertino *et al.* pointed out that "*mechanisms for enforcing access control policies* *based on data contents*" are needed for comprehensive data protection. More relevant to the proposed research, the notion of *content-based access control* has been used in relational access control specification [39], [40], multimedia database [41], [42], web 2.0 [43]–[45], and digital libraries [46], etc. However, their definition of "content" is quite different from ours. In particular, in [39], [40], [46], [47] the notion of *content* refers to attribute values or definitive concepts extracted from digital library objects. Access privileges are *statically* specified based on relationships between user credentials and attributes/concepts. Similarly, policy-based access control models [28], [29], [48] bind access rights with user credentials, however, the decision is still based on definitive values of the attributes (e.g. users with `title`="`physician`" could access patient records in his/her `department`). In [41], RBAC is extended to specify access control policies on image content (captured as attributes). [49] and [42] enforce access control of video databases based on text annotations on videos, while [50] manages videos in clusters (based on visual content), and supports more flexible access control. In all cases, explicit and static rules are required – user credentials, video content and access control policies are all explicitly defined *a priori*.

More recently, [43]–[45] enforces access control in Web 2.0 based on tags of messages, where tags are learned from the message content. Access control is explicitly specified on tags, for instance, there are explicit rules such as: "`[family members]` are allowed to access messages tagged with `[home]`". To handle the dynamics in modern enterprise applications, a few recent proposals attempt to infer access control provisioning from known decisions using supervised learning, when a decision cannot be directly made from available policies [51], [52]. This approach is effective when a good number of training samples (known access decisions) are available, and training and testing samples statistically follow the same distribution. On the other hand, concept-level access control has also been proposed for the semantic web [53]. Last, the terms *context* and *semantic* has been used in various access control approaches. *Context* mostly refers to the operational context of the user [54], while *semantic* is often used to indicate the semantic of data schema and access control policies, especially in data integration and federation applications [55], [56].

Our notion of content-based access control is significantly different from existing approaches, we refer to the *semantic* content semantic similarity of data in RDBMS or XML DB, as well as the notion of *approximation* and *implicit access control specification*. In our approach, *content* refers to the meanings of data objects. Last, Oracle's CONTEXT index is essentially an *inverted index* for text retrieval, which is very different from the *context* used in access control literature.

VII. Discussions and future work

Computational complexity. The efficiency and scalability issues are major matrices in evaluating access control approaches. As we have shown, CBAC could be efficiently enforced using native functions from Oracle, especially with CONTEXT indexing, blocking and labeling. Meanwhile, we would also give a formal analysis of computational complexity, which is purely theoretical, and does not consider any DBMS optimization.

First, without any indexing and blocking, the DBMS needs to perform a pairwise comparison between every record and the seed records, to make access control decisions. The computational complexity is $O(N \cdot m)$, where N denotes the total number of records (usually very large), and m denotes the size of the base set. With vector space model, the complexity for each comparison would be $O(T)$, where T denotes the dimensionality of the term space. The computation for comparison could be easily reduced to $O(t)$, where t being the number of distinct terms in the user's base set. Hence, the overall computational complexity for a query would be $O(N \cdot m \cdot t)$. As we see, query processing time is linear to the number of records, while m and d are relatively small.

Next, with blocking, the DBMS first selects x clusters of records from the total c clusters, and then perform pairwise comparison between seed records and records within the x clusters. Assuming that the size of clusters are relatively balanced, there will be N/c records in each cluster on average. The computation for selecting top x clusters is $O(c \cdot m \cdot d)$, while the computation for enforcing CBAC for records within the x clusters would be $O((N/c) \cdot m \cdot d \cdot x)$. Hence, the total computation would be:

$$O(c \cdot m \cdot d + \frac{N}{c} \cdot m \cdot d \cdot x) \geq O(2m \cdot d\sqrt{N \cdot x})$$

Hence, the blocking mechanism reduces the overall computation to $O(m \cdot d \cdot \sqrt{N \cdot x})$. It could be reduced to $O(m \cdot d \cdot \log{(N \cdot x)})$, with multi-level blocking.

Negative rules and conflict resolution. In database access control, negative rules are employed to disallow the user to access specified records. Usually, positive rules allow the user to access a (relatively large) set of records, while negative rules exclude particular records from the set. Negative rules could be supported in CBAC, for instance, to specify that "Agent Alice cannot access records similar to case X" in Example 1. To enforce negative rules, another set of seeds and policies are generated to exclude the selected records, i.e., to represent the semantics of "NOT (similar to X) ". Meanwhile, in the case of conflict rules (e.g. a positive rule grants access to a record, while a negative rule forbids it), the negative rule usually takes precedence. In some access control models, the rule with a smaller scope takes precedence. In CBAC, we also have the capability to specify that the rules with higher content-similarity take precedence.

Advanced user and content modeling. Finally yet importantly, the CBAC model provides no restrictions on user and content modeling. We have presented a proof-of-concept implementation of the CBAC model with vector space and topic-based models. In practice, more complicated user and content modeling methods could be employed. For instance, it will be helpful to include advanced content models such as

Latent Semantic Indexing [57], opinion extraction [58], and sentiment analysis [59]. However, understanding the semantic content of unstructured text content is a very difficult problem, which is outside of the scope of this paper. It is one of the main tasks of our future work.

Record similarity assessment. In this paper, the access control verdict made by CBAC is purely based on the content similarity between the base set and the requested records. Moreover, other mechanisms could be used to further exploit the internal relationships between records. For instance, if two records often appear together in the same base set, it implies some implicit relationships between the records, although they could be very different in terms of semantic content. We are investigating advanced record understanding mechanisms, beyond content-based modeling.

VIII. Conclusion

In this paper, we introduce CBAC model and enforcement mechanisms. As a complement to the conventional access control approaches, the CBAC model is most suitable for content-centric information sharing, when content plays a major role in access decision-making, and approximation is allowed by application. CBAC is a two-phase model: in the initial authorization, a *base set* is identified for each user. CBAC then automatically extends the base set utilizing the semantic similarity between the base set and the requested records. We formally present CBAC model, and demonstrate an enforcement mechanism of this model on Oracle VPD. Meanwhile, to improve the computational efficiency of the enforcement mechanism, we introduce an offline similarity assessment approach, and a blocking approach. We further improve the accuracy of semantic content matching with a tagging mechanism. Experimental results show that the access control decisions made by CBAC are reasonable, and the overhead is acceptable.

Acknowledgment

The work has been in part supported by NSF CNS-1422206, NSF OIA-1028098, and the University of Kansas General Research Fund GRF-230142.

The authors would like to thank anonymous reviewers for their valuable feedback.

References

[1] A. Appari and M. E. Johnson, "Information security and privacy in healthcare: current state of research," *International journal of Internet and enterprise management*, vol. 6, no. 4, pp. 279–314, 2010.

[2] B. Malin and E. Airoldi, "Confidentiality preserving audits of electronic medical record access," *Studies in health technology and informatics*, vol. 129, no. 1, p. 320, 2007.

[3] A. A. Boxwala, J. Kim, J. M. Grillo, and L. Ohno-Machado, "Using statistical and machine learning to help institutions detect suspicious access to electronic health records," *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 498–505, 2011.

[4] L. Rostad and O. Edsberg, "A study of access control requirements for healthcare systems based on audit trails from access logs," in *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*. IEEE, 2006, pp. 175–186.

[5] A. Ferreira, R. Cruz-Correia, L. Antunes, P. Farinha, E. Oliveira-Palhares, D. W. Chadwick, and A. Costa-Pereira, "How to break access control in a controlled manner," in *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*. IEEE, 2006, pp. 847–854.

[6] D. M. Sharma, "On the role of nlp in linguistics," in *Workshop on NLP and Linguistics*, 2010.

[7] C. Wu and Y. Chen, "A survey of researches on the application of natural language processing in internet public opinion monitor," in *Computer Science and Service System,*, june 2011.

[8] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[9] M. J. Pazzani and A. Meyers, "Nsf research awards abstracts 1990-2003." [Online]. Available: http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html

[10] C. Hoare, "Algorithm 65: Find," *Commu. ACM*, vol. 4, 1961.

[11] P. Ferragina and U. Scaiella, "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)," in *ACM CIKM*, 2010.

[12] S. Jajodia and R. Sandhu, "Toward a multilevel secure relational data model," in *ACM SIGMOD*, May 1990.

[13] M. Winslett, K. Smith, and X. Qian, ""Formal Query Languages for Secure Relational Databases"," *ACM Trans. on Database Systems*, vol. 19, no. 4, pp. 626–662, 1994.

[14] J. Moffett, M. Sloman, and K. Twidle, "Specifying discretionary access control policy for distributed systems," *Computer Communications*, vol. 13, no. 9, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0140366490900085

[15] R. K. Thomas and R. S. Sandhu, "Discretionary access control in object-oriented databases: Issues and research directions," in *16th National Computer Security Conference*, 1993, pp. 63–74.

[16] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[17] P. P. Griffiths and B. W. Wade, ""An Authorization Mechanism for a Relational Database System"," *ACM Trans. on Database Systems*, vol. 1, no. 3, pp. 242–255, Sep. 1976.

[18] E. Bertino, L. M. Haas, and B. G. Lindsay, "View management in distributed data base systems," in *VLDB*, 1983, pp. 376–378.

[19] E. Bertino and L. Haas, "Views and security in distributed database management systems," in *EDBT*, 1988, vol. 303, pp. 155–169.

[20] P. Samarati and S. de Vimercati, "Access control: Policies, models, and mechanisms," in *Foundations of Security Analysis and Design*, 2001, vol. 2171, pp. 137–196.

[21] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino, "A unified framework for enforcing multiple access control policies," in *ACM SIGMOD*, 1997, pp. 474–485. [Online]. Available: http://doi.acm.org/10.1145/253260.253364

[22] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati, "A temporal access control mechanism for database systems," *IEEE TKDE*, vol. 8, no. 1, pp. 67 –80, feb 1996.

[23] E. Bertino, P. A. Bonatti, and E. Ferrari, "Trbac: A temporal role-based access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 191–233, Aug. 2001. [Online]. Available: http://doi.acm.org/10.1145/501978.501979

[24] M. Winslett, N. Ching, V. Jones, and I. Slepchin, "Using digital credentials on the world wide web," *J. Comput. Secur.*, vol. 5, no. 3, pp. 255–267, Jun. 1997. [Online]. Available: http://dl.acm.org/citation.cfm?id=353686.353691

[25] E. Bertino, S. Castano, and E. Ferrari, "Securing XML Documents with AuthorX," *IEEE Internet Computing*, vol. 5, no. 3, pp. 21–31, 2001.

[26] R. Krishnan, R. Sandhu, J. Niu, and W. H. Winsborough, "Foundations for group-centric secure information sharing models," in *ACM SACMAT*, 2009. [Online]. Available: http://doi.acm.org/10.1145/1542207.1542227

[27] J.-W. Byun and N. Li, "Purpose based access control for privacy protection in relational database systems," *The VLDB Journal*, vol. 17, pp. 603–619, 2008.

[28] L. Kagal, T. Finin, and A. Joshi, "A policy based approach to security for the semantic web," in *The Semantic Web - ISWC*, 2003, pp. 402–418. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39718-2_26

[29] R. Bhatti, D. Sanz, E. Bertino, and A. Ghafoor, "A policy-based authorization framework for web services: Integrating xgtrbac and ws-policy," *IEEE Intl. Conf. on Web Services*, pp. 447–454, 2007.

[30] B. Hicks, S. Rueda, D. King, T. Moyer, J. Schiffman, Y. Sreenivasan, P. McDaniel, and T. Jaeger, "An architecture for enforcing end-to-end

access control over web applications," in *ACM SACMAT*, 2010, pp. 163–172. [Online]. Available: http://doi.acm.org/10.1145/1809842.1809870

[31] J. S. Park, R. Sandhu, and G.-J. Ahn, "Role-based access control on the web," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 1, pp. 37–71, Feb. 2001. [Online]. Available: http://doi.acm.org/10.1145/383775.383777

[32] E. Bertino, B. Catania, E. Ferrari, and P. Perlasca, "A logical framework for reasoning about access control models," *ACM TISSEC*, vol. 6, no. 1, pp. 71–127, Feb. 2003.

[33] H. Takabi and J. B. Joshi, "Stateminer: an efficient similarity-based approach for optimal mining of role hierarchy," in *ACM SACMAT*, 2010, pp. 55–64. [Online]. Available: http://doi.acm.org/10.1145/1809842.1809853

[34] I. Molloy, N. Li, Y. A. Qi, J. Lobo, and L. Dickens, "Mining roles with noisy data," in *ACM SACMAT*, 2010, pp. 45–54. [Online]. Available: http://doi.acm.org/10.1145/1809842.1809852

[35] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin, "Access control policy combining: theory meets practice," in *ACM SACMAT*, 2009.

[36] P. Rao, D. Lin, E. Bertino, N. Li, and J. Lobo, "An algebra for fine-grained integration of xacml policies," in *ACM SACMAT*, 2009, pp. 63–72. [Online]. Available: http://doi.acm.org/10.1145/1542207.1542218

[37] V. Rao and T. Jaeger, "Dynamic mandatory access control for multiple stakeholders," in *ACM SACMAT*, 2009, pp. 53–62. [Online]. Available: http://doi.acm.org/10.1145/1542207.1542217

[38] E. Bertino, G. Ghinita, and A. Kamra, *Access Control for Databases: Concepts and Systems*. Now Publishers Inc, 2011, vol. 3, no. 1-2.

[39] E. Bertino, P. Samarati, and S. Jajodia, "An extended authorization model for relational databases," *IEEE TKDE*, vol. 9, no. 1, pp. 85 –101, jan/feb 1997.

[40] L. Giuri and P. Iglio, "Role templates for content-based access control," in *Proceedings of the second ACM workshop on Role-based access control*, ser. RBAC '97, 1997, pp. 153–159. [Online]. Available: http://doi.acm.org/10.1145/266741.266773

[41] S. K. Tzelepi, D. K. Koukopoulos, and G. Pangalos, "A flexible content and context-based access control model for multimedia medical image database systems," in *Workshop on Multimedia and security: new challenges*, 2001, pp. 52–55.

[42] N. A. Tran and T. K. Dang, "A novel approach to fine-grained content-based access control for video databases," in *Database and Expert Systems Applications, 2007. DEXA '07. 18th International Workshop on*, 2007, pp. 334–338.

[43] M. Hart, R. Johnson, and A. Stent, "More content-less control: Access control in the web 2.0," *IEEE Web 2.0 Privacy and Security Workshop*, 2007.

[44] S. Monte, "Access control based on content," TKK Technical Reports in Computer Science and Engineering, B. TKK-CSE-B10. http://www.cse.tkk.fi/en/publications/B/10/papers/Monte_final.pdf, 2010.

[45] M. A. Hart, "Content-based access control," http://udini.proquest.com/view/content-based-access-control-pqid:2402941941/, 2011.

[46] N. Adam, V. Atluri, E. Bertino, and E. Ferrari, "A content-based authorization model for digital libraries," *IEEE TKDE*, vol. 14, no. 2, pp. 296 –315, mar/apr 2002.

[47] H. Amjad, "A context aware content based federated access control system for healthcare domain," ECS Masters Thesis, Purdue University. http://docs.lib.purdue.edu/ecetheses/13/, 2007.

[48] P. Reddivari, T. Finin, and A. Joshi, "Policy-based access control for an rdf store," in *Proceedings of the Policy Management for the Web workshop*, 2005, pp. 78–83.

[49] E. Bertino, M. A. Hammad, W. G. Aref, and A. K. Elmagarmid, "An access control model for video database systems," in *Proceedings of the ninth international conference on Information and knowledge management*, ser. CIKM '00. New York, NY, USA: ACM, 2000, pp. 336–343. [Online]. Available: http://doi.acm.org/10.1145/354756.354838

[50] E. Bertino, J. Fan, E. Ferrari, M.-S. Hacid, A. K. Elmagarmid, and X. Zhu, "A hierarchical access control model for video database systems," *ACM Trans. Inf. Syst.*, vol. 21, no. 2, pp. 155–191, Apr. 2003. [Online]. Available: http://doi.acm.org/10.1145/763693.763695

[51] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proceedings of the Second ACM CODASPY*, 2012, pp. 157–168. [Online]. Available: http://doi.acm.org/10.1145/2133601.2133622

[52] Q. Ni, J. Lobo, S. Calo, P. Rohatgi, and E. Bertino, "Automating role-based provisioning by learning from examples," in *ACM SACMAT*, 2009, pp. 75–84. [Online]. Available: http://doi.acm.org/10.1145/1542207.1542222

[53] L. Qin and V. Atluri, "Concept-level access control for the semantic web," in *ACM workshop on XML security*, 2003, pp. 94–103. [Online]. Available: http://doi.acm.org/10.1145/968559.968575

[54] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila, "A semantic context-aware access control framework for secure collaborations in pervasive computing environments," in *The Semantic Web - ISWC 2006*, ser. Lecture Notes in Computer Science, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, Eds. Springer Berlin Heidelberg, 2006, vol. 4273, pp. 473–486.

[55] C.-C. Pan, P. Mitra, and P. Liu, "Semantic access control for information interoperation," in *SACMAT*, 2006, pp. 237–246. [Online]. Available: http://doi.acm.org/10.1145/1133058.1133091

[56] B. Fabian, S. Kunz, M. Konnegen, S. Mller, and O. Gnther, "Access control for semantic data federations in industrial product-lifecycle management," *Computers in Industry*, vol. 63, no. 9, pp. 930 – 940, 2012.

[57] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '99, 1999, pp. 50–57. [Online]. Available: http://doi.acm.org/10.1145/312624.312649

[58] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, "Opinion extraction, summarization and tracking in news and blog corpora." in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, vol. 100107, 2006.

[59] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.