An Access Control Model for Video Database Systems*

Elisa Bertino† [1] Moustafa A. Hammad [2] Walid G. Aref [2] Ahmed K. Elmagarmid [2]

[1] Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41 20135 Milano, Italy
bertino@dsi.unimi.it

[2] Department of Computer Sciences,
Purdue University,
West Lafayette IN 47907-1398
{mhammad,aref,ake}@cs.purdue.edu

ABSTRACT

A novel approach for modeling access control in video databases is presented. The proposed access control mechanism uses both the semantics and the structural composition of video data. The unit of authorization, a video element, can either be a sequence of video frames or a video object that appears as part of a frame, e.g., the face of an anonymous person in an interview. The components of the access control model are the video elements, the potential users, and the mode of operation, e.g., viewing, or editing. Video elements are specified either explicitly by their identifiers or implicitly by their semantic contents, while users are characterized by the user credentials. An algorithm is presented that determines the authorized portions of a video that a given user may acquire, given the user's credentials, the video content descriptions, and the type of requested video operations. The description of the implementation of a prototype MPEG-2 based video database system with access control are also presented.

1. INTRODUCTION

Handling video data needs multidisciplinary skills and leads to problems that go far beyond video storage and retrieval or image processing. Video possesses unique characteristics such as volume and complexity of data even for simple and short video clips.

A lot of work [7, 16] has been done in describing video data content. In general, video content includes *audio-visual* content, that specifies audio signal, color intensity and distribution, texture patterns, object motions, to name a few, and *semantic* content, that deals with the knowledge or information contained in a given video segment. However, little work has been done in providing a secure and organized access to video data. Video access is naturally described in terms of its semantic contents, for example, in a rating system, violent movies are restricted to audience above certain age, based on video violent *content*. We address in this paper how to build video access control based on both *video semantic contents and users credentials* which is a more flexible and natural way to express authorization for video data. In a conventional database environment access control is usually performed against a set of authorizations stated by security administrators according to some security policies. In its most basic form, an authorization is specified as a triple $< s, o, m >$, where subject s is authorized to access object o under mode m, where the mode refers to the actions that can be executed on the protected object, such as read or write. Such an approach must be properly extended in order to satisfy the additional challenging requirements characterizing video database (VD) systems.

Video data is used in a variety of applications environments, such as medical applications, teaching, environmental protection, manufacturing processes, scientific research, just to name a few. In such environments, it is often the case that different classes of users within the same organization must receive different authorizations for the same set of data. For example, consider a school giving access to VD to both teachers and students. Consider a set of videos illustrating firearms. Whereas teachers can be allowed to see all such videos, the students may only be allowed to see the videos not showing how to operate guns with the exception of students having age equal or greater than 18. There is thus the need for models and mechanisms supporting the specification of authorizations on the basis of user qualifications or characteristics rather than user identity.

Another crucial requirement is to support content-dependent authorizations on video data objects. By content-dependent authorizations we mean that authorizations are granted or denied to a given user (or class of users) depending on the actual content of the video data objects. Consider again the firearms example, supporting this requires determining which videos show gun operations. Such a requirement thus calls for the integration of the access control mechanism with mechanisms able to express and model semantic contents of video data.

A third important requirement results from the fact that video data has a hierarchical structure, for example video stream, video scenes, frames and video objects. This requirement calls for an access control model supporting vary-

*This work is supported by the National Science Foundation under grants 9972883-EIA, 9974255-IIS and 9983249-EIA; Indiana 21st Century grant from the State of Indiana; and grants from HP, IBM, Intel, Telcordia and CERIAS.

†The work of this author was funded by grant from CERIAS.

CIKM 2000, McLean, VA USA
© ACM 2000 1-58113-320-0/00/11 . . .$5.00

Figure 1: Video clips with blurred parts (face of a person)

ing granularity levels of authorized objects. For example, Figure 1 gives a set of frames in a video clip where the face of the person is blurred, for the purpose of hiding identity.
In this paper we propose an access control model satisfying the above requirements. Specific features of the proposed model include: access control specification for video data objects based on their semantic contents rather than their identifiers; flexible specification of authorization based on the notion of user credentials; varying granularity of authorized objects ranging from an entire video, to part of a video to specific portions of the frames. Our model also provides functions for resource usage controls, such as limitations on the play time or video resolution. We propose a video data model that captures the compositional structure of video data and can easily represent the semantic contents. We have integrated our access control with the proposed video data model.
The rest of this paper is structured as follows. Section 2 introduces related work and contrast it with our work. Section 3 introduces the video data model and the video elements used throughout this paper. The video authorization model and the detailed specification of its components are described in Section 4. Section 5 presents the access control mechanisms and the algorithm proposed in this paper. Section 6 introduces the access control architecture and our prototype implementation. Section 7 gives some concluding remarks.

2. RELATED WORK

Several efforts have been reported to extend conventional database access control models to deal with new data types and to provide new functions in authorization management. Such efforts include temporal authorization models [3], and extended authorization models for relational databases [4]. Such models are not however fully adequate for the protection of information in a video database system. The main reason is that authorizations are specified in terms of user, or user groups, and object identifiers rather than in terms of user profiles and object contents. Also, to the authors' knowledge, supporting varying levels of protection granularity for video database objects has not been addressed before. The only approach we are aware of has been proposed by Kumar and Babu [13]. This approach only allows one to hide entire frames for specific classes of users and has no support for sub-frame restriction. Moreover, even though such an approach considers users as partitioned into user categories, it does not support authorizations containing predicates against user profiles.

In [1]a content-based access control for textual data objects in digital libraries has been recently proposed. That approach supports authorizations to be associated either with an entire document or with parts of it. Such a model has no provision for video access control for the following reasons. Video data has a more articulated object granularity, e.g., entire video, sequence of frames, and parts of frames. Moreover, spatio-temporal properties of video data should be considered in describing the access context, e.g., some video elements may be restricted in a specific context that is described using spatial or temporal relations between their contents.
Content-dependent access control has been addressed both in relational DBMS and in object-oriented databases, through the use of views [10]. Content-based access control is enforced by simply specifying some conditions against attribute values of data objects. In contrast, due to the nature of video data objects, content-dependent access control for a VD must be based on the semantics of the video objects, rather than on the attributes characterizing them. Video attributes often only deal with physical characteristics of the video data objects (for example, the color intensity and distribution, texture patterns, and number of frames or segments composing the video) and therefore are not significant for access control. Also, the spatio-temporal properties and the varying granularities of video data should be considered in constructing a view of authorized data. For example, view definition should be able to describe situations like, restricting appearance of a person in a specific context.

3. THE UNDERLYING VIDEO DATA MODEL

In this section, we introduce the video model used in the development of our access control model. In general, the video model should be able to represent the video semantic contents, provide a way to represent the spatio-temporal property of video data, provide a way to model different granularity of video data and represent the compositional structure of video. For example, one view of video data is as a collection of sequences, a sequence is a collection of shots and a shot is a contiguous set of frames that represent a continuous action in time and space [6].
We model video data by three major elements; *a video stream, a video segment, and a video object.* Refer to Figure 2 for an illustration of the video data model. A video stream represents the continuous sequence of frames that forms the video context. It is associated with textual annotations that semantically describe the video element. The annotation can be added manually or automatically extracted from video (e.g. capturing closed captioning associated with video). A video segment represents a sequence of frames that are interrelated. The relationships among the frames are user defined. The relationships may be built on physical characteristics (e.g., when generated by scene change detection techniques [12]), or on a semantic view (e.g., the sequence of frames where a certain actor appears or that contains related shots in a movie). Each video segment has textual annotations that describe its semantics.
The third element of the video model is the video object that represents a visual video object (e.g., a human face, a car, a building, etc.). A video object may have many *video object occurrences.* Each video object occurrence has spatio-temporal features attached with it that describe its geometry, e.g., a sequence of polygons or minimum bound-

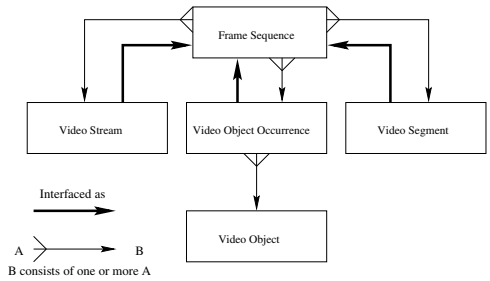

Figure 2: Main components of the video data model.

ing rectangles (MBRs) representing the boundaries of the video object over a period of time. Both, the video object and the video object occurrence have textual annotations that give the necessary semantic contents.

Each one of the video elements has a unique identifier and is regarded as a new sequence of frames, even though it may be composed of different video elements. For a video object, it is composed of the set of video object occurrences. For the other video elements, the set of pairs (video element id, [start frame, end frame]) represents the compositional information.

The video data model allows a video editor to compose a new video element from existing video elements, e.g., composing a news clip, and supports the representation of multiple video granularities, e.g., a whole video stream, a video segment and a video object. In our model each of the video elements can either be materialized (has a physical existence in a raw video data) or virtually represented, if composed from other existing video elements.

4. AN AUTHORIZATION MODEL FOR VIDEO DATABASES

Integrating access control into database management systems is usually achieved by specifying a set of *authorization rules* and *control procedures* [5]. Authorization rules describe *who* is allowed to access *what* in the database. Control procedures deploy these rules on database transactions.

In a video database context, the general form of an authorization rule is a 3-tuple <*subject, object, mode*>, where the object refers to a video element as defined in Section 3. The reader should not confuse the term object with the term video object also defined in Section 3. In the following sections we present a detailed specification of the main components of the video database authorization rules, i.e., the subject, the object and the mode.

4.1 Subject Specification

Subjects in an authorization model represent entities trying to access the database. In our model, we will assume subjects to be end-users. However, our model can be easily extended by including subjects, such as roles and groups [15]. A suitable access control model should be able to incorporate information about user characteristics and profiles while describing authorization to video. To this purpose, we adopt in our model the notion of credentials [1]. A credential is a set of security-relevant information, called *credential attributes*, pertaining to a given user. The credential mechanism allows one to specify subjects in authorization rules not only by using their user-ids (or other system-defined identification mechanism), but also implicitly by specifying the conditions users need to verify in order to access a given video, or sets of videos. The credential mechanism provides a language for formulating *credential expressions*, that is, a Boolean combination of predicates, against which user characteristics are matched. Evaluation of a credential expression thus results in a set of users that satisfy the expression requirements. To ease the process of credential specification, credentials with similar structures are grouped into a *credential type*.

In the following, we recall from [1] definitions of some of the above-mentioned notions. In the following definitions, we assume that the following sets are given: $\mathcal{AN}$ - a set of attribute names; $\mathcal{T}$ - the set of the possible types (such as `integer`, `real`, `Boolean`, `character`, and `string`) of attributes in $\mathcal{AN}$; $\mathcal{V}$ - the set of legal values for types in $\mathcal{T}$. Moreover, we denote the set of credential-type identifiers and the set of credential identifiers with $\mathcal{CT}$ and $\mathcal{CI}$, respectively. We use $\mathcal{U}$ to denote a set of user identifiers and we use $A(ct\_id)$ to denote the set of the names of attributes in credential type ct_id, where $ct\_id \in \mathcal{CT}$.

DEFINITION 4.1. **(Credential Type)** *[1] A credential type is a pair (ct_id, attr), where ct_id* $\in \mathcal{CT}$ *is the credential type identifier; and attr is a set containing an item for each attribute of the credential type. attr in turn is a triple (name, dom, a_type), where name* $\in \mathcal{AN}$ *is the attribute name, dom* $\in \mathcal{T}$ *is the attribute domain, and a_type* $\in$ {`opt, mand`} *specifies whether the attribute is optional (*`opt`*) or mandatory (*`mand`*).* □

Example: The following is an example of credential type: (Student, {(address, string, mand), (GPA, number, mand), (status, string, opt), (registered, Boolean, mand)}) △

A credential is an instance of a credential type and provides the corresponding values to the specified attributes, as specified by the following definition.

DEFINITION 4.2. **(Credential)** *[1] A credential c is a 4-tuple (c_id, user_id, state, ct_id), where c_id* $\in \mathcal{CI}$ *is the credential identifier, user_id* $\in \mathcal{U}$ *is the identifier of the user with whom the credential is associated; state* $= (a_1 : v_1, .., a_n : v_n)$*, where* $a_1, \ldots, a_n \in A(ct\_id)$ *are the names of the attributes of c, and* $v_1, \ldots, v_n \in \mathcal{V}$ *are their values; and ct_id* $\in \mathcal{CT}$ *is the identifier of the credential type of which c is an instance.* □

Example: The following is an example of credential: (c_1, John, (address:Waldron street, GPA:3.5, Status:Graduate, Registered:Yes), Student). △

Note that the same subject can have different credentials. Credential expressions are specified by a simple language which consists of a set of variables Var_U, ranging over the set $\mathcal{U}$ of user identifiers and a set of predicate symbols $Pred$ of arity one, with type Var_U. For each credential type $ct \in \mathcal{CT}$, a corresponding predicate symbol $ct()$ is defined in $Pred$. Such predicates capture the associations of users with credential types. Also, predicates can be defined as a distinct set of user identifiers.

Expressions that can be specified in our language are formally defined as follows.

DEFINITION 4.3. **(Credential Expression)** *Let $\Theta = \{=, \neq, >, <, \geq, \leq, \in, \notin, \subseteq, \not\subseteq, \subset, \not\subset, \supset, \supseteq, \not\supset, \not\supseteq\}$ be a set of relational comparison operators. The set $\mathcal{CE}$ of credential expressions is built from atoms and Θ as follows. Atoms can be of the following types:*
• *$P(x)$, where $P \in Pred$ and $x \in Var_U$;*
• *$x.a$ op v, where $x \in Var_U$, $a \in \mathcal{AN}$, $v \in \mathcal{V}$, and op $\in \Theta$.*
Then the set $\mathcal{CE}$ of credential expressions is recursively defined as follows:
• *Every atom is a credential expression.*
• *If CE_1 and CE_2 are credential expressions, then $CE_1 \wedge CE_2$, $CE_1 \vee CE_2$, $\neg CE_1$, (CE_1) are also credential expressions.* □

The evaluation of a given credential against a credential expression consists of replacing the attribute names, in the credential expression, with the corresponding values in the credential and determining the truth value of the resulting credential expression. The set of subjects to which authorizations apply can thus be specified by means of some credential expression, as stated by the following definition.

DEFINITION 4.4. **(Subject Set Specification)** *A subject set is a credential expression in $\mathcal{CE}$.* □

Example: In what follows we present some examples of the different forms that can be used to introduce subjects in our model.
• `(Viewer(x) ∧ (x.age ≥ 18))`: this expression denotes all viewers who are ≥ 18 years old.
• $\{\texttt{uid}_1, \texttt{uid}_2, \texttt{uid}_3 ...\}$(x): this is an explicitly specified list of authorized subjects. △

Credential types can be extended to include audio and visual characteristics of subjects that affect the way they access video contents. For example, a speak impaired person may be allowed to view video while replacing the associated audio content by a superimposed sign language (introduced by an illustrator) and hence save the bandwidth and speed up the display process.

4.2 Object Specification

As discussed in Section 3, a video element can be represented as either a video stream, a video segment or a video object. The specification of an *authorization object* in our model is based on these video elements.
Video elements can be specified either directly, by providing their identifiers, or indirectly through their *contents*. For example, the semantic contents, the audio-visual contents, etc. Contents can be combined in *content expressions*. A content expression involves one or more video contents combined according to some Boolean, spatial, temporal or spatio-temporal operators. Temporal relations are defined over the interval between the start frame and end frame, of the video elements and cover all the possible temporal relations between two intervals (e.g., *overlap, during, start, end, meet (and their inverses), before, after and equal*) [2]. The spatial relations apply only on video objects in a frame

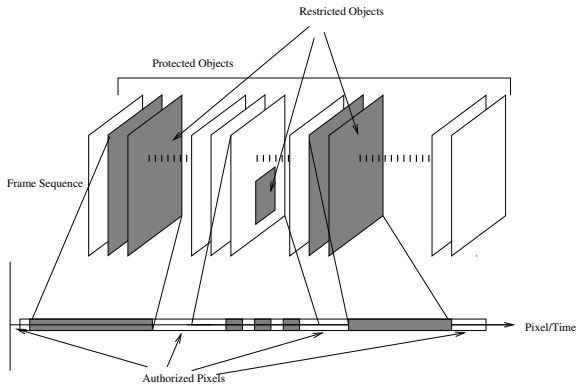

Figure 3: The relationship between authorized, protected and restricted objects.

and can either be topological relations (e.g., *overlap, disjoint, touch and inside*), directional relations (e.g., *north, south, east, west, above, below, left, right, middle, center*), or distance relations (e.g., *far, near, close*). Spatio-temporal relations are also defined among video objects. For example, the *approach* relation indicates an approaching video object to another video object and is interpreted as the spatial distance between them decreases over their common life time interval [11]. Content expressions are specified by a language similar to the one used in credential expression. In the following definition we use the following sets: Var_V - a set of variables ranging over the video elements identifiers; $Pred$ - a set of predicate symbols of arity one, with type Var_V; $\mathcal{CN}$ - a set of video contents, for example contents may represent the annotation associated with each video element; $\mathcal{VCN}$ - a set of values for video content; $\mathcal{VOP}$ - a set of video operators (spatial, temporal, and spatio-temporal operators).
Content expressions that can be specified in our language are formally defined as follows.

DEFINITION 4.5. **(Content Expression)** *Let Θ be a set of operators defined over the video contents (e.g. Θ can be the contain operator, to denote if a certain keyword(s) exists in a video element annotation, or Θ can be a comparison operator). The set $\mathcal{CNE}$ of content expressions is built from atoms which can be of the following types:*
• *$P(x)$, where $P \in Pred$ and $x \in Var_V$;*
• *$x.c\ \Theta\ v$, where $x \in Var_V$, $c \in \mathcal{CN}$, $v \in \mathcal{VCN}$.*
Then the set $\mathcal{CNE}$ of content expressions is recursively defined as follows:
• *Every atom is a content expression.*
• *If CNE_1 and CNE_2 are content expressions and op $\in \mathcal{VOP}$, then $CNE_1 \wedge CNE_2$, $CNE_1 \vee CNE_2$, $\neg CNE_1$, (CNE_1), CNE_1 op CNE_2 are also content expressions.* □

The evaluation of a content expression will result in those video elements with contents satisfying the expression. The following example shows a content expression, where the content of interest is the video annotation, *annot*, associated with video elements.

Table 1: Definition of $\ominus$ operator on video elements

$\ominus$	Vst_{ro}	Vsg_{ro}	Vo_{ro}
Vst_{po}	The set of frame sequences $\in Vst_{po}$ and $\notin Vst_{ro}$	The set of frame sequences $\in Vst_{po}$ and $\notin Vsg_{ro}$	Vst_{po} with blurred appearance of Vo_{ro}
Vsg_{po}	The set of frame sequences $\in Vsg_{po}$ and $\notin Vst_{ro}$	The set of frame sequences $\in Vsg_{po}$ and $\notin Vsg_{ro}$	Vsg_{po} with blurred appearance of Vo_{ro}
Vo_{po}	The set of frame sequences including Vo_{po} and $\notin Vst_{ro}$	The set of frame sequences including Vo_{po} and $\notin Vsg_{ro}$	The set of frame sequences including Vo_{po} with blurred appearance of Vo_{ro}

Vst, Vsg, Vo denote video stream, video segment and video object, respectively, and the subscripts *po, ro* refer to protected and restricted element.

Example: The content expression
(x.annot contain 'Charles De Gaulle') *DURING* (y.annot contain 'World War II') will get all video elements (x) that contain the phrase 'Charles De Gaulle' in their annotations and that temporally fall during the period of video elements (y) that contain the phrase 'World War II' in their annotations. △

Objects in our authorization model are introduced by specifying two components. The first component identifies a video element, or set of video elements (that is, a video stream, a video segment, a video object or sets of them) the user wants to access. We refer to this component as a *protected object set*. But video is more informative in terms of its contents, and even though the user may have access to video, he may be restricted to access part(s) of it. So, the second component, in our specification specifies *censored* parts, where the user should be denied access to. Those censored parts represent video elements that should not be accessed by the user. We refer to those censored parts as *restricted object set*. The set of objects finally obtained after excluding the censored parts is referred to as the *authorized object set*. Those are the actual objects to which the authorization applies. Figure 3 shows the relationship between protected, restricted, and authorized objects with the mapping on pixel/time domain, where each frame can be considered as a two dimensional array of pixels.
A key feature of our model is that both the protected and the restricted object sets are specified according to the same language. Therefore, a seamless integration between the specification of the two components is achieved in the specification of the authorized object set. We refer to the specification of a protected or restricted object set as *object set specification*. A formal definition of object specification, protected or restricted, is given in what follows.

DEFINITION 4.6. **(Object Set Specification)** *The object set is defined as a content expression.* □

Note that the object specifications can be quite heterogeneous, ranging from specifications given just in terms of an explicit list of video elements to specifications given by imposing conditions on the video contents, according to the content expression language. However, the actual objects denoted by the specification are all video elements. More specifically:
• If the objects are explicitly specified by a list of video elements, such as video streams, video segments or video objects, then these video elements represent the object set specification.
• If the objects are specified through a content expression, then the video elements with contents satisfying the expression represent the object set specification.
The following definition states our notion of authorized object set.

DEFINITION 4.7. **(Authorized Object Set Specification)** *Let po and ro be object set specifications defined according to Definition 4.6. An authorized object set specification, aos is a two-component expression po.ro, where the first component is the protected object set, and the second component is the restricted object set.* □

Note that the restricted object set component, *ro*, is optional, i.e., a missing *ro* part indicates an empty set, $\emptyset$, of restricted objects.
To fully define our approach, we need to state what is the actual semantics of an authorized object set specification. The semantics formally states how we determine the actual objects denoted by a given authorized object set specification.

DEFINITION 4.8. **(Semantics of Authorized Object Set Specification)** *Let aos = po.ro be an authorized object set specification defined according to Definition 4.7. The semantics of aos is defined by the following expression:*
$aos = po \ominus ro$,
where the $\ominus$ operator is defined in Table 1. □

Example: The following are some examples of authorized object set specifications.
• po = (x.annot contain "Drug addiction interview")
  ro = {PersonID}(y)
This specification denotes all video elements reporting interviews about drug addiction without, however, showing the addicted person. `PersonID` in the specification is the video object identifier of the addicted person and denotes the restricted object in the video.
• po = (x.annot contain "Firearms")
  ro = (y.annot contain "gun" DURING y.annot contain "operate")
This specification denotes all videos showing firearms, by restricting the video portions that show how actually guns are operated. △

4.3 Mode Specification

Low level operations, such as physical read and write operations, are not semantically meaningful for access control in video database. Therefore, in our model we introduce a set of abstract operations that are relevant to the way users actually access VD objects.
Users of VD go through the following stages. The user first submits a request for a given video. The VD server processes the request, and returns to the user either the annotations associated with the video, or a list of representative

Table 2: Video privileges, VP, provided by the access control model

Class	Privilege	Meaning
View	annotation	To display the results as the associated annotations only. It speeds up the query response time.
	RFrames	To display the results as a set of RFrames. Displaying only RFrames allows one to return relevant information about the result and also save the bandwidth by not returning the whole video.
Play	(period, quality)	To play video element. *period* specifies permission to play for a specified period of time, say for 10 minutes; $\star$ indicates playing the entire video. *Quality* specifies the desired displaying quality of video (low | $high$) Play only indicates unlimited period and High quality.
Edit	annotation	To edit the annotations of the video elements.
	video	To modify, delete or add video elements to VD.

frames, RFrames. The user can then request to play the video or submit a more detailed request based on the intermediate result. Also, the authorization may permit playing for a specified duration (such as the first 10 minutes of the video), or playing the video data elements according to different resolutions [17]. The last function uses authorization to control resource usage. Suitable access modes should be devised corresponding to operations performed in such a scenario. Another group of operations for which access control should also be provided include operations for editing the video elements or for introducing new ones. In general, authorizations to perform such operations should be given to few, selected users. The highest privilege operations are to add, delete or modify the existing video elements since such operations may greatly affect other video elements.
The different modes of operation, *video access privileges*, that are provided as part of our model are described in Table 2. The privileges in the table can be ordered, in terms of increasing power, according to user preference. In other words the model permits user to indicate which operation is subsumed by the other, for example viewing the annotation may be considered more serious than viewing the RFrames, if the text will reveal more information than the RFrames do. The $\prec_p$ is used to represent a total order relation between video privileges, for example one possible order of privileges is: View(annotation) $\prec_p$ View (RFrames) $\prec_p$ Play (period, quality) $\prec_p$ Edit(annotation) $\prec_p$ Edit(video).

4.4 Video Authorization Rule Specification

In this section a formal definition of authorization rules is introduced. In the definition we assume that the following sets are given: $\mathcal{P}$ - a set of time intervals expressed according to some time unit; $\mathcal{Q}$ - a set of quality levels.

DEFINITION 4.9. **(Authorization Rule)** *Let s be a subject set specification defined according to Definition 4.4. Let aos be an authorized object set specification defined according to Definition 4.7. Let m be an access mode in the set* {`view(annotation)`, `view(RFrames)`, `edit(annotations)`, `edit(video)`} $\bigcup$ {`play (`p_i`,` q_i`)` | $p_i \in \mathcal{P}$, $q_i \in \mathcal{Q}$}. *An authorization rule is defined as the tuple (s, aos, m).* □

According to the above definition, an authorization rule has the following components:
• s: it is a set of authorized subjects. Subjects are introduced as credential expression and s includes all subjects that satisfy this expression.
• *aos*: it is a set of authorized video elements representing the difference between two sets calculated according to the $\ominus$ operator. The elements of both sets are specified by content expression as stated in Definition 4.6.
• m: it is the video operation allowed for the subjects on the specified objects. If m is the `play` mode, the authorization may optionally contain a time duration and a quality level.
In what follows, we present several examples illustrating the features of our authorization model.

Example: Let *VstID* be the identifier of the interview video stream, and *VoID* be the video object representing the face of the interviewed person. Then, authorization
AR1 =(s: Viewer(x) AND (x.Class=General Audience),
aos: (po: VstID(y), ro: VoID(z)),
m: Play)
gives the play authorization on the TV-interview, while hiding the face of the interviewed person, to all viewers whose class is general audience. △

Example: The authorization
AR2 =(s: Student(x) AND (x.major = History),
aos: (po:(y.annot contain "World War II") AND
(y.annot contain "documentary movies")),
m: Play)
gives the play authorization for browsing the World War II video library to all college students whose major is history. △

5. ACCESS CONTROL

The main role of the access control mechanism is to verify that user u, trying to access object o, using a privilege p, is authorized to do so. This access control scenario is quite general [5]. However, video access control has the following distinct features. First, subjects are introduced by using credential expressions. Hence, the *subject verification process* should consider the satisfaction of user credential to the provided credential expression. Also, the *object verification process* should consider both possibilities of existence of object identifier or satisfaction of content expression by the contents of the requested object. In addition, authorized objects are specified as two components, protected and restricted objects. These components are not necessarily temporally or spatially disjoint. For example, a protected object may represent the video stream and a restricted object may represent certain video segments or even video objects in this video stream. This requires the access control to apply some filtering based on the restriction part, such as clipping restricted frames, or obscuring or blurring restricted subframes.
The access control algorithm is specified in Figure 4. The algorithm works as follows: It checks each authorization rule in $\mathcal{AR}$ set that has u as one of its subjects and such that p is an allowed video operation. Functions *IsSubject and IsMode* perform these checks. The previous checking step is important in the evaluation of access control since many requests can be denied at this point without the overhead of object retrieval and access checking. In our model we allow

ALGORITHM 5.1. **Access Control Algorithm**

INPUT: *[1] An access request (u, o, p), [2] The authorization rules set $\mathcal{AR}$*
OUTPUT: *[1]* ACCEPT *and return o', [2]* REJECT *otherwise*
METHOD:

```
relevant_ar_set:=∅
For each ar(s, aos, m) ∈ AR do
    If ( IsSubject*(u, s) ∧ IsMode†( p, m) ) Then
        relevant_ar_set ← relevant_ar_set ∪{ar}
    EndIf
EndFor
If ( relevant_ar_set ≠ ∅ )Then
    o' ← o
    For each ar ∈ relevant_ar_set do
        o' ← objects that belong to both o' and aos
    EndFor
    If (o' ≠ ∅ )Then
        return(ACCEPT, o')
    Else
        return(REJECT)
    EndIf
Else
    return(REJECT)
EndIf
```

* *IsSubject(u,s) returns TRUE if user satisfies subject expression, else returns FALSE.*
† *IsMode(p,m) returns TRUE if $p \prec_p m$ or p=m, else returns FALSE.*

Figure 4: Access control algorithm

the subject to have many authorization rules and we always follow the most conservative ones. Hence, several authorization rules can satisfy those conditions and all of them are collected in *relevant_ar_set*. If after searching the whole $\mathcal{AR}$ set, relevant_ar_set is empty then the user request is rejected and the check is complete. Otherwise, for each authorization rule in relevant_ar_set the requested video element is restricted to the authorized object set. After processing all the relevant_ar_set, o' will contain the authorized objects. If o' is empty, this means the user's request is rejected, otherwise it is accepted and returns o'.

6. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The system architecture is depicted in Figure 5. The *authorization manager* is responsible for the full management of both the *authorization rules base* and *credentials base*. Through the authorization manager, one can add, modify, or delete user credentials or authorization rules. *The access control manager* implements the access control algorithm specified in Section 5. The *content manager* processes content expressions and returns the video elements with contents satisfying the expression. The *video data manager* is responsible for handling of video data (video stream, video segment and video object). The access control manager also communicates with the *filtering effects manager* to perform any necessary exclusion of frames or blurring of certain subframes.

We have implemented our access control model and tested the functionalities described in this paper. We use video data sources in MPEG-2 format. MPEG stores video data in a hierarchical structure of sequences, GOP (group of pictures) and pictures [14]. An MPEG sequence consists of

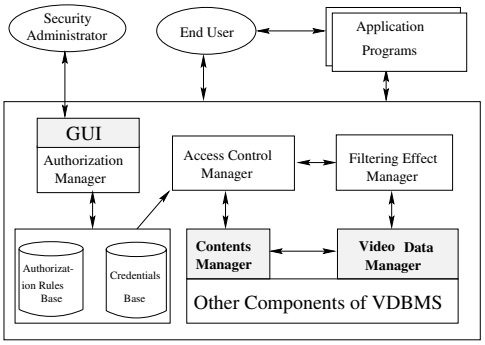

Figure 5: System architecture for a secure video database management system

one or more GOPs. We build an index table on top of each MPEG file to be able to address GOP and sequences directly. We use a relational database management system to implement our video model as described in Section 3 and to store all necessary data (authorization rules, credentials and index tables of MPEG source files).

The security administrator is the only user who has the rights to introduce new authorizations. The administrator should specify the mode of operation, the subject (as a credential expression) and both the protected and restricted object sets (as a content expression). The system pre-evaluates the content expressions to retrieve the corresponding video elements and store their identifiers. The pre-computation of the protected and restricted objects is important in order to speed up the computation process as will be discussed later. The system also allows the administrator to introduce new users and define their profiles.

We have implemented a parser to process content expressions, introduced in Definition 4.6. The parsing output is a set of SQL commands that access the underlying relational video model. The SQL commands are then executed to retrieve the target video elements. All video operations (Boolean, temporal, spatial, or spatio-temporal) introduced in Section 4.2 are easily expressed in SQL syntax. For example, temporal operations are specified as Boolean checks against *start* and *end* frames of video elements. Also, spatial operations are expressed using Boolean checks against the points representing the geometric description of the video objects.

We deploy the following strategy in executing a user's request. The request is directed only to those video elements that belong to a *protected object set*. When obtaining the result (the set of video elements that satisfy the user's request and also belongs to the protected object set), we restrict them with the restricted object set. The restriction process follows Definition 4.8, and may involve clipping of frames or blurring of video objects.

Clipping of frames is achieved by mapping the result to a *non overlapping* sequence of frames and comparing them with a *non overlapping* version of restricted frame sequences, then clipping sequences that fall in the restricted region. This is implemented by one scan over the sorted video intervals as shown in Figure 6. Blurring parts of frames is accomplished by following the same steps as clipping but instead of removing the restricted frames, we process (blur) video objects in the intersected frames based on the stored ob-

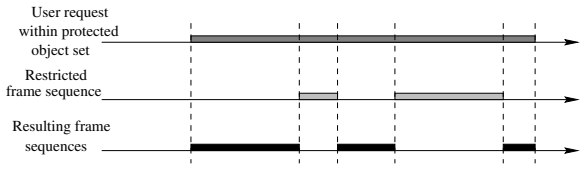

Figure 6: Clipping of video frames.

ject geometric description. Clipping of frames is performed in two steps, the first step is a logical operation, where all frames clipping is performed on the stored interval information of video elements. The second step is performed only as the user requests to play the video and involves streaming the processed intervals to the user for display. Only at this step we need to deal with raw video stream. The blurring process is also done in two steps, where the first step only records the need to blur a video object (identified by a unique id) and the frame intervals to apply the changes. The second step involves accessing the pictures that contain the video objects and then blurring the area specified by the minimum bounding rectangle (MBR). This approach has the advantages of speeding up the video access process (all data operations are performed against the video meta data, in the data model) and accesses to raw video data is delayed until display time. Our technique also does not slow down the display process significantly since, video is usually compressed and we integrate the process of filtering in the decoding process.

7. CONCLUSION AND FUTURE WORK

In this paper we presented an access control model for video databases. The main components of authorization rules have been defined in video context. Our model allows one to specify authorization subjects by their identifiers or according to their credentials. The model also provides a clear definition of authorized objects that can be specified implicitly by using content expressions. A distinct contribution of our work is to provide access control for different video granularities ranging from a whole video stream to sub-frame regions or video objects. In order to implement these techniques, filtering effects are incorporated into the access control mechanism. Filtering effects are used to hide a sequence of frames, or to blur sub-frame regions in these sequences. They can be extended to deal with audio and text as well. The access model also provides a categorization of privileges that are meaningful for video data. The privileges we have devised are abstract and suitable to interact with video. They range from the privilege of just viewing the annotations associated with video to full control on video elements. Because our authorization model is based on video contents, it can be easily applied to different video data models. This is a relevant feature of our work since several efforts are currently on-going in the area of video content description (e.g. MPEG-7 [9]).

8. REFERENCES

[1] N. Adam, V. Atluri, E. Bertino, and E. Ferrari. A content-based authorization model for digital libraries. *IEEE Trans. on Knowledge and Data Engineering, to appear.*

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM.*, 26(11):832–843, November 1983.

[3] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Trans. on Database Systems, 23(3):231-285*, 1998.

[4] E. Bertino, P. Samarati, and S. Jajodia. An extended authorization model. *IEEE Trans. on Knowledge and Data Engineering, 9(1):85-101*, 1997.

[5] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security.* Addison-Wesley, 1995.

[6] G. Davenport, T. Aguierre Smith, and Natalio Pincever. Cinematic primitives for multimedia. *IEEE Computer Graphics and Applications Magazine, July*, 1991.

[7] A. K. Elmagarmid and et al. *Video Database Systems. Issues, Products and Applications.* Kluwer Academic Publishers, 1997.

[8] E. Fernandez, E. Gudes, and H. Song. A model for evaluation and administration of security in object-oriented databases. *IEEE Transactions on Knowledge and Data Engineering, 6(2): 275-292*, 1994.

[9] Requirement Group. Mpeg-7 context and objectives. *International Organization of Standardization, ISO/IEC JTC1/SC29/WG11. Coding of Moving Pictures and Audio*, 1999.

[10] E. Gudes, E. Fernandez, and H. Song. Evaluation of negative, predicate and instance-based authorizations in object-oriented databases. *In Database Security, IV: Status and Prospects, Elsevier publ*, 1991.

[11] H. Jiang and A. K. Elmagarmid. Spatial and temporal content-based access to hypervideo databases. *The VLDB Journal*, 7(4):226–238, December 1998.

[12] H. Jiang, A. Helal, A. K. Elmagarmid, and A. Joshi. Scene change detection techniques for video database systems. *Multimedia Systems.*, 6(3):186–195, May 1998.

[13] P. S. Kumar and G. P. Babu. Intelligent multimedia data: data + indices + inference. *Multimedia Systems.*, 6(6):395–407, December 1998.

[14] John L. Mitchell, William B. Pennebaker, and Didier J. LeGall. *MPEG Video Compression Standard.* Chapman and Hall, 1997.

[15] R. sandhu et Al. Role-based access control models. *IEEE Computer, pages 38-47, February*, 1996.

[16] V. S. Subrahmanian. *Principles of Multimedia Database Systems.* Morgan Kaufmann, 1997.

[17] K. Yamaashi, Y. Kawamata, M. Tani, and H. Matsumoto. User-centered video: Transmitting video images based on the user's interest. *in Proc. Computer Human Interaction. Denver, Colorado*, 1995.