Access Control, Confidentiality and Privacy for Video Surveillance Databases

Bhavani Thuraisingham
Department of Computer Science
University of Texas at Dallas
bxt043000@utdallas.edu

Gal Lavee
Department of Computer Science
University of Texas at Dallas
gxl034000@utdallas.edu

Elisa Bertino
Department of Computer Science
Purdue University
bertino@cerias.purdue.edu

Jianping Fan
Department of Computer Science
UNC-Charlotte
jfan@uncc.edu

Latifur Khan
Department of Computer Science
University of Texas at Dallas
lkhan@utdallas.edu

ABSTRACT
In this paper we have addressed confidentiality and privacy for video surveillance databases. First we discussed our overall approach for suspicious event detection. Next we discussed an access control model and accedes control algorithms for confidentiality. Finally we discuss privacy preserving video surveillance. Our goal is build a comprehensive system that can detect suspicious events, ensure confidentiality as well as privacy.

Categories and Subject Descriptors
K.6 [**Management of Computing and Information Systems**], K.6.5 [**Security and Protection**].

General Terms
Security.

Keywords
Access Control, Confidentiality, Privacy, Video and Surveillance.

1. INTRODUCTION
Due to their benefits on public safety and crime-fighting, surveillance cameras have been used in public places such as airports, harbors, and subway stations [10, 1, 2]. Fear of terrorism and the availability of ever-cheaper surveillance cameras have accelerated this trend [16, 6]. In order to enhance information gathering and extraction of significant knowledge, it is critical to enable sharing and integration of surveillance videos from multiple organizations and groups, so that early detection of the terrorists' movements, activities and associations, or of other objects of interest, can be effectively achieved by analyzing large-scale databases of surveillance videos. In addition to count-terrorism and law enforcement applications, analysis of surveillance videos also has many applications in Command and Control. For example, an organization belonging to a coalition may monitor a specific region and analyze the surveillance data of the war fighter as well as the adversary captured in that region. There is an urgent need for multiple organizations of the coalition to share the surveillance data so that the big picture is obtained of the combat operation.

An important problem in this context is related to the fact that confidentiality requirements of organizations owning the video databases and privacy regulations may require that the integration of such video databases be executed so that confidentiality of sensitive information can be preserved. In this context it is important to notice that each organization contributing its own video databases may have its own guidelines and policies concerning what objects in a video are to be considered privacy sensitive. An organization may be willing to share its video databases with other organizations as long as confidentiality of privacy-sensitive objects is assured. Our effort will focus on security-preserving data sharing. That is, we will develop techniques for ensuring the confidentiality of the privacy sensitive information contained in the surveillance data.

In addition to ensuring confidentiality and privacy, we desire a method to efficiently restrict access to video surveillance data. Surveillance video data by its very nature can contain information that is sensitive from a security (national or otherwise) as well as a privacy perspective. Of course, we must acknowledge that the truly sensitive information lies in the high-level semantic content of a video data object and not in the low-level representation of the information. In other words we are not concerned with restricting the data based on its color, texture or shape content but rather by the human interpretation of the content of the video sequence.

To this end we propose a model for authorization objects in a surveillance video database. This model allows for the definition of the various components of video data including semantic (meaningful to humans) events and objects as well as other

SACMAT'06, June 7–9, 2006, Lake Tahoe, California, USA.

information such as physical location and timestamp. To allow these admittedly subjective and broad (fuzzy) human concepts we have to define a set of ontology hierarchies into which our concepts fit. This allows us to relate concepts to each other. Ultimately this model allows us to define access control authorizations on video objects based on all aspects of video data.

In this paper we will first describe the grammar that serves as the building blocks for this access control model. We will then describe how video objects are represented within our model, including discussion about semantic concept component hierarchies. Authorization subjects and their representation will also be considered along with credential expressions allowing the "query-like" specification of these subjects. After this we will discuss the specification of our authorization policy and the differences between explicit and derived authorizations. Derived authorizations take advantage of the extensive use of hierarchical structures throughout our access control model to expand our explicit policy base. Finally we will discuss the access control algorithm which reconciles user requests with the authorization policy base to determine whether to grant or deny access to the requested object.

The organization of this paper is as follows: In section 2 we discuss related work for caves control of video and surveillance data. In section 3 we will provide an overview of our project on suspicious event direction. Our access control model and algorithm, which is the main focus of this paper, will be discussed in section 4. In section 5 we will discuss some of the privacy and confidentiality issues. The paper is concluded in section 6.

2. RELATED WORK

As mentioned in section 1, our main focus in this paper is our access control model and algorithm for surveillance databases. Our model allows us to restrict access to surveillance video data based on semantic content. Related work in access control has offered many notions that can be applied to such a model.

Early work focused on access control for multimedia databases [17]. More recently there has been work on security for geospatial databases. Atluri et al [3] discuss an authorization model for geospatial data such as satellite images and digital vector data such as maps and geographical overlay information. Like our data domain, this kind of data has the unique property of being tied to specific real-world geographic coordinates as well as a real-world timestamp indicating the time the image was captured. The model utilizes this information in the representation of authorization objects. An object type is also part of this representation. Examples of object types in this authorization model are: Geospatial object, image/raster, Satellite image, IKONOS image, Aerial Photo, vector, point, line, etc… These example types come from closed set of available types. The system defines these types and their relationships in a hierarchical taxonomy. A policy authorization takes the form of a tuple: <oid, type, latitude, longitude, height, width, timestamp, resolution, linked data>. This format describes the real-world geospatial coordinates of the object (as a minimum bounding box) as well as the time the data was captured and other information such as image resolution and relational data tied to the image. For querying these types of objects the paper defines literals and geospatial and temporal terms, along with operations that can be performed on these terms. The paper also extends the concept of subject specifications with the concept of subject credentials. These credentials are specified as instantiations of credential types in a credential type hierarchical taxonomy. These credential types are made more robust by their ability to contain temporal and geospatial terms describing the subject. Using the subject and object definitions we can now specify robust access control policy authorizations as tuples containing a subject credential expression , an object expression , a privilege mode and a temporal term indicating the point in time where this authorization is valid. Extended privilege modes pertaining to geospatial data such as zoom-in, overlay, identify, animate, download, etc… are defined by the authorization model. These are also defined in a hierarchical fashion. The extensive use of hierarchical taxonomies allows a much broader base of authorization policy to be deduced from a fewer number of explicit policy authorizations. The subject, object, and privilege hierarchy can be used to derive the majority of the policy base from these explicit authorizations defined by the user.

Our proposed access control model for surveillance video data is based in part on the concepts in this paper. Like geospatial data our data domain is tied to real-world set of geospatial coordinates and a real-world timestamp. We can additionally make use of the object type and credential type hierarchies (modified to represent our domain). In our model we extend the representation of the authorization objects to include sets of the two possible categories of semantic content in a video surveillance object: events and objects. Each of these sets can contain multiple entries, but each entry will come from a closed set of concepts whose relationship is defined by a hierarchical taxonomy. This will enable a system allowing very robust expression of access control policy as well as the ability to deduce many derived authorizations from a few explicit ones.

Bertino et al [5] propose an access control model for video database systems. In this paper the semantic structure of the video is defined by a human video editor. The semantics of the video data can be described in terms of three categories: video streams, video segments and video objects. A video stream is defined as the series of frames that describes the semantic context. It is associated with an annotation and can therefore be derived "automatically" using closed-captioning streams. A video segment relates several sequences of frames using user-defined semantic relationships. A video object describes a semantic object in the scene such as a person, a car, or a tree. It can then be instantiated into a video object occurrence and associated with geometric properties that represent the appearance of the object in the scene. Each frame sequence objects is then associated with one or more of these semantic descriptors. Authorization subjects are specified using a credential scheme similar to the one suggested in this paper using credential types and their instantiation.

This paper considers a substantially larger domain of data then does our model which focuses on surveillance video applications. In this paper the semantic concepts are drawn from the infinite set of human describable ideas. This works well for a small video databases relatively invariant domain where all concepts can be consistently labeled the same way, but is limited in its accuracy since the definition of what constitutes a semantic event is sensitive to the subjective interpretation of its human labeler. That being said, we do not in much detail discuss an objective way to

label a video sequence with semantics provided we have a closed set of hierarchically related semantic concepts.

The video object and their corresponding video object occurrences can be likened to our semantic video objects. However, the video objects in this paper are drawn from an arbitrary infinite set subject to the human labeler's interpretation at the time of the video labeling. Furthermore, objects are not related in any way to one another, so it is not possible to automatically derive authorization policy based on parent/child concept relationships, as is done in this paper using the semantic object type hierarchy. Parallel to our concept of "semantic events", defined as the behavior of an entire sequence is the notion of video streams in this paper. Video streams, relate a manual or automatic (based on closed captioning) annotation, to a continuous series of frames. Again, the concepts enclosed in this annotation are drawn from an infinite set of concepts that are difficult to relate to one another. Our semantic event hierarchy, by contrast, enables us to annotate our scenes (which can be considered a series of continuous frames) with a closed set of semantic concepts to eliminate ambiguity. It also relates semantic events to one another so we can generalize authorization policy at a later time. Finally, our paper also expands on the concept of credential types in this paper by organizing these in a hierarchical taxonomy that allows further implicit policy derivation.

3. OUR PROJECT ON SUSPICIOUS EVENT DETECTION

In order to enable this model we need to define a way to automatically derive semantic annotation from low-level information. Our previous work [12] proposes method to perform such a derivation with regard to semantic event information. We propose to model an event as the amount of low-level feature change (our low-level feature of choice is the color intensity value) within a scene. We can then learn common behavior within a surveillance system, give each of these common even events a semantic label, and then use these label to classify future video sequences into semantic categories. With a well defined event hierarchy in place we can relate these objects to one another for the purposes of data visualization, retrieval, and of course access control.

In this section we will provide some background information on the overall project which is suspicious event detection from surveillance data. Our system takes a new unlabeled video sequence and multiple labeled video sequences representing different types of possible events. It produces a visualization of the content in the unlabeled video sequence as output to the user. The user can adapt this visualization according to their preferences (i.e. what type of event they consider to be suspicious) using the Video Analysis Tool interface. Figure 1 shows a block diagram of the system design.

The new video sequence is read in and stored as a matrix of RGB values over time (width*height*3*number of frames). This phase can be thought of as the extraction of low-level features. An event representation (see below) for several overlapping subsequences is generated for use in event detection. These newly generated events are compared to a set of predefined events using the event comparison (distance) function defined below. The events are then classified by use of the nearest neighbor algorithm.

Once each of the overlapping events has a label the system generates a summary of events contained within the new video sequences and encodes it in an XML document. This document is used in conjunction with user input to provide the appropriate visualization of the video content.

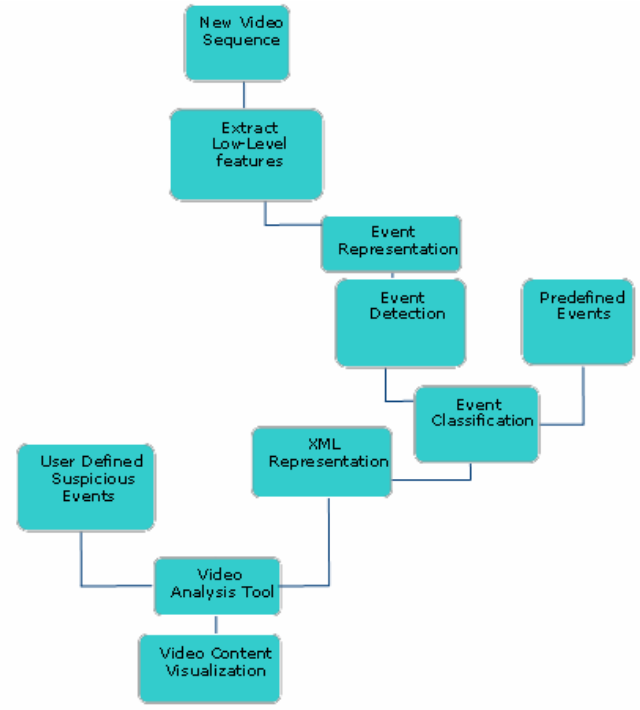

Figure 1: System Design

Essentially the functions we have designed and implemented include event representation, event extraction, and event comparison and event detection; In addition we also carry out XML video annotation. The availability of machine-readable annotation documents, whether these are in XML or a specialized video content ontology language format, is a big step towards the bridging of the semantic gap. A video analysis tool that takes this kind of annotation as input and organizes the corresponding video segment accordingly is certainly conceivable. This kind of utility could function as an aid to a surveillance analyst searching for "Suspicious" events within a stream of video data.

This activity of interest may be defined dynamically by the analyst during the running of the utility. The definition would then be compared to the video annotation and similar frames could be flagged for further analysis. Alternately, the analyst could define what is considered to be normal behavior and any annotated event deviating from this norm would be flagged.

The analyst would then consider all flagged areas to determine if action is required. A color coated scroll bar (with video segments of interest indicated by different colors) may be used as a graphical interface for this task. The analyst may then further refine the search parameters by marking false and true positives. Details of our system and experiments are given in [13].

4. THE MODEL

4.1 Grammar

We define a mathematical notation that allows formal representation of our access control subjects, objects and authorizations. This formalism makes heavy use of the conventions of set theory and builds on authorization models described in [11, 4].

4.1.1 Literals

A literal $l \in L = \{N \cup A \cup G \cup T\}$ where *N* is the set of all natural numbers, *A* the set of all strings, *G* the set of all geospatial data types, and *T* the set of all temporal data types.

4.1.2 Temporal Data Types

A temporal data type $T = \{T' \cup \overline{T} \cup \{now, UC\} \cup TT\}$ where T' is the set of discrete time points on a continuous time-line which can be mapped to the set of natural numbers with total order among each $t \in T'$. $\overline{T}$ is the set of time intervals of the form $\bar{t} = [t_b, t_e]$ indicating the set of all timepoints $t \in T'$ where $t_b \le t \le t_e$. $\{now, UC\}$ is a symbolic timepoint pairing where now represents the current point in time and UC (Until change) represents the future timepoint $t \in T'$ when some change is to occur. $TT = \{TC \cup TV\}$ is the set of temporal terms where TC is the set of constant time points and $TV = \{t_i, t_b, t_e, now, UC\}$ is the set of temporal variables. t_i represents a variable for any discrete timepoint. t_b and t_e represent variables for the beginning and end points of a temporal interval. The now variable represents a current timepoint . UC is a variable representing a timepoint when change occurs. $TF = \{+, -, *, /\}$ is the set of all temporal functions. If $tt_1, tt_2 \in TT$ and $f \in TF$ then $tt_1 f\ tt_2 \in TT$. That is if we compose two temporal terms using a temporal function the result is also a temporal term.

4.1.3 Geospatial Data Types

A geospatial data type $G = \{GC \cup GT\}$ where GC is the set of all geospatial constants and GT is the set of all Geospatial terms. $GC = \{\Psi \cup \Lambda \cup \Omega\}$ where Ψ is the set of geographic points such that each $\psi \in \Psi$ is represented by the pair $< lt, \lg >$ where lt represents the earth's latitude and *lg* represents the earth's longitude using a standard geospatial coordinate system. Λ is the set of geographic lines such that each $\lambda \in \Lambda$ is a pair of points $< \psi_1, \psi_2 >$ where $\psi_1, \psi_2 \in \Psi$. Additionally the representation of each line $\lambda \in \Lambda$ contains the pair $< lt, \lg >$ representing the latitude and longitude of the center point of the line on the earth's coordinate system. Ω is the set of regions where each $\omega \in \Omega$ is a closed polygon representing a region defined by a series of points $< \psi_1, \psi_2, \psi_3, ..., \psi_n >$ where $\psi_1, \psi_2, \psi_3, ..., \psi_n \in \Psi$ and $\psi_1 = \psi_n$. Additionally, each $\omega \in \Omega$ contains the 4-value tuple $< lt, \lg, h, w >$ describing the minimum bounding box which will engulf the polygon. *lt* and *lg* represent the earth coordinate latitude and longitude of the center of the bounding box while *h* and *w* represent the height and width of the box, respectively. $GT = \{GC \cup GV\}$ where GC is the set of geospatial constants as defined above and $GV = \{address, region, area, place\}$ is the set of geospatial variables. *GF={union, difference, intersect ,xor}* is the set of all Geospatial functions. If $gt_1, gt_2 \in GT$ and $\mu \in GF$ then $gt_1 \mu\ gt_2 \in GT$. That is if we compose two geospatial terms using a geospatial function the result is also a geospatial term.

4.1.4 Operators

We define several types of operators to be used with our literals. Specialized operators exist to operate on special data types. We define $LOP = \{=, \neq, <, >, \le, \ge\}$ to be the set of logical operators, *GOP={contain, equal, overlap, meet}* to be the set of geospatial operators, *TOP={before, after, during}* to be the set of temporal operators and CH*OP={contain, containparent}* to be the set of component hierarchy operators.

4.2 Authorization Objects

Our authorization objects, the actual video data to which we wish to restrict access, are represented in the form of a 7 value tuple. This tuple contains information about the content of a particular video object. Some of this content information pertains to high-level semantic information such as events and objects. This information is stored as a set of concepts taken from a "closed-world" hierarchical taxonomy which relates these concepts to one another. Other content information such as location and timestamp is represented as a special data type that allows more meaningful specification of this unique kind of content.

Formally, each surveillance video object $v \in V$ is represented by the tuple *v=<oid, type, events, objects ,location, timestamp>* where *oid* is a unique object identifier, *type* is the type of the surveillance video object, *events* is the set of all semantic events contained within the video, *objects* is the set of all semantically meaningful objects contained within the video, *location* is a geospatial term describing the point or region in which the video data was captured in terms of a physical location(i.e. tied to earth's coordinate system and *timestamp* is a temporal term describing the timepoint or interval in which the video data was captured .

Each surveillance video object is of an object *type* defined in the surveillance video object type hierarchy. The surveillance video object type hierarchy *S* is comprised of the set of object types related by subtype relationships. We use the notation $ot_i \prec_{OT} ot_j$ to denote that ot_i is a subtype of ot_j. Surveillance objects can be moving video or still image capture and they can be from a specific class of location. These categories

and subcategories of video objects are an example of surveillance object type relationships that can be defined using a hierarchical type taxonomy. Because of the subtype relationship between the object types we may visualize the object type hierarchy as a tree. An example of this hierarchy is shown in Figure 2.

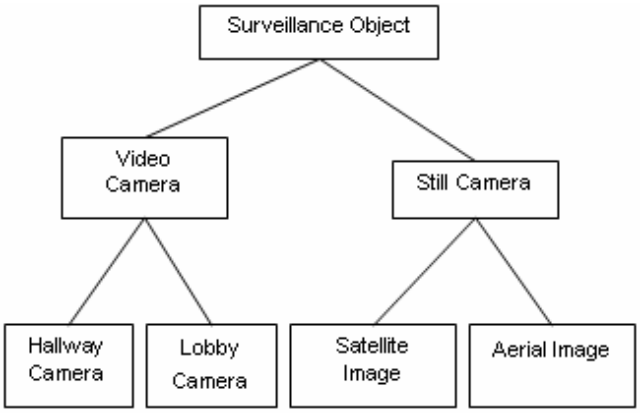

Figure 2: An Example of the Surveillance Video Type Hierarchy

Events is the set of semantic events occurring within the video object. Members of this set are drawn from the event type taxonomy (another hierarchical description of event types and their relationships). This set may contain any combination of the event types contained in the event type taxonomy (and only from this taxonomy) or contain the empty set.

Objects is the set of semantic objects contained within the video object. Members of this set are drawn from the semantic object type taxonomy (a hierarchical description of object event types and their relationships). This set may contain one or more objects from the object type hierarchy in any combination (multiple members of the same object type may also comprise a legal object set) or be empty.

Location is the term indicating the geographic earth coordinates of where the surveillance video object was captured. We can use geospatial terms defined above to specify this information as well as a gazetteer service to translate real world place names to such terms.

Timestamp is the term describing the real world time when the video was captured. We use temporal terms which include timepoints and time intervals to specify the values for this attribute.

4.3 Component Hierarchies

The hierarchical taxonomy approach (figures 3 and 4) used to define the surveillance video object types is a convenient way to represent relationships between semantic concepts. This simple structure can be surprisingly effective in modeling the complex relationships humans attribute to different concepts. For example in the object hierarchy knowing that a ball concept is a subtype of the toy concept allows us to relate authorizations referring to the "toy" class of objects to video objects containing the "ball" object. We utilize this representation method to define our closed set of object and event concepts and how they relate to one another. We refer to these hierarchical taxonomies representing semantic concepts as the component hierarchies.

The semantic video event type hierarchy is a set of all possible event types and their relationship (parent/child) to one another. Formally, the semantic video event hierarchy *SVEH* is comprised of a set of semantic video event types *SET*= $\{set_1, set_2, \ldots\}$ which are related by subtype relationships. We use the notation $set_i \prec_{SET} set_j$ to denote that set_i is a subtype of set_j.

Similarly, the semantic video object hierarchy is a set of all possible semantic object types and their relationships to one another. Formally, the semantic video object hierarchy *SVOH* is comprised of a set of semantic video event types *SOT*= $\{sot_1, sot_2, \ldots\}$ which are related by subtype relationships. We use the notation $sot_i \prec_{SOT} sot_j$ to denote that sot_i is a subtype of sot_j.

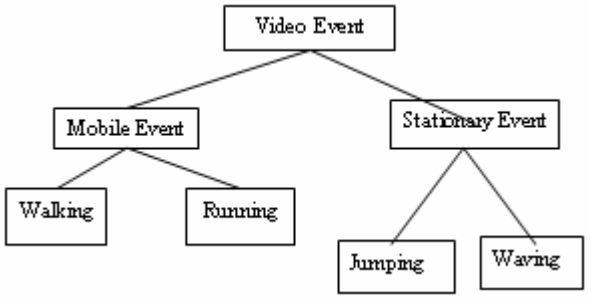

Figure 3: An example of the Semantic Event Type Hierarchy

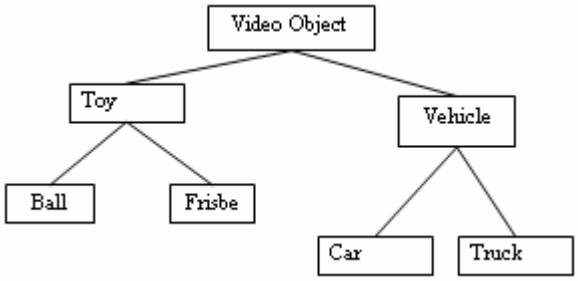

Figure 4: An example of the Semantic Object Type Hierarchy

4.4 Video Object Expressions

Video object expressions describe the object for which access control is to be applied. These expressions are expanded and made more robust so that a video object may be specified not only by its object ID but rather by any of its attributes or their combination. This is similar to querying a relational database using a complex SQL query specifying a particular set of records.

We use access functions to reference the different components of our surveillance video objects for use in our expressions. *AF = {type(x), rectangle(x), timestamp(x), objects(x), events(x)}* is the set of these access functions where *x* is an object variable. *type(x)* returns the surveillance video object type of object *x.* This returned type will be a member of the surveillance video object type hierarchy discussed above. *rectangle(x)* returns a minimal bounding box describing the geospatial coordinates of the physical location contained within the video object. *timestamp(x)* returns the temporal interval in absolute time during which the video object occurs. *Objects(x)* returns the set of semantic objects contained in video object *x* comprised of types defined in the

semantic object type hierarchy. *Events(x)* returns the set of semantic events contained in object *x*. This set contains events with types drawn from the event type hierarchy.

oid(x) is a video object expression if *oid* is an object identifier. This object expression references a specific video object by its index. *Type(x)=y* is an object expression if $y \in OT$ the set of all video object types defined in the video object type hierarchy. This expression references all objects of a specific video object type. *rectangle(x) gop gt* is a video object expression if $gt \in GT$, the set of all geospatial terms and $gop \in GOP$, the set of all geospatial operators. *timestamp(x) top tt* is a video object expression if $tt \in TT$, the set of all temporal terms and $top \in TOP$, the set of all temporal operators. *Events(x) chop y* is a video object expression if $y \in SVEH$, the set of all semantic video event types defined in the semantic video event type hierarchy and $chop \in CHOP$, the set of all component hierarchy operators. *Objects(x) chop y* is a video object expression if $y \in SVOH$, the set of all semantic video objects defined in the semantic video object type hierarchy and $chop \in CHOP$, the set of all component hierarchy operators. Video object expressions may also be compound combinations of these simpler forms of video object expressions joined by boolean operators.

Examples of video object expressions take the following form:

1. *Events(x) contains "running" ^ timestamp(x) after 5:00 PM*

2. *Objects(x) contains "Toy" ^ Type(x)="Video Camera"*

4.5 Authorization Subjects

A discussion of an access control model would not be complete without mention of subject specifications. In our proposed model we use the concept of user credentials to authorize users. That is, each user entity, in addition to having a unique user id or belonging to a group also possesses a set of credentials. Each credential is an instantiation of a certain credential type, the template for credentials in which the set of credential attributes, and whether they are optional or obligatory is defined. Specific values are assigned to these attributes when a new user instantiates the credential type. A subject may instantiate any number of credential types. These credential types are defined in a credential type hierarchy relating each credential type to the other credential types (figure 5).

Formally, the credential type hierarchy *CTH* is comprised of the set of all credential types *CT*= $\{ct_1, ct_2, ct_3 .....\}$ which are related by subtype relationships . We use $ct_i \prec_{CT} ct_j$ to denote that ct_i is a subtype of ct_j .

Credential type templates are specified in terms of their attributes. These attributes may be of several data types including temporal and geospatial data. In the credential type definition we also specify whether the specification of this attribute is mandatory or optional. These templates are later instantiated for subject credentials. Formally each credential type *ct=< id, A>* where id is

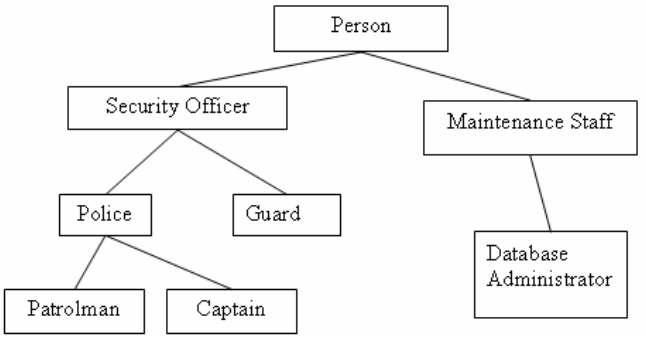

Figure 5: Example of Credential Type Hierarchy

the unique identifier of the credential type and $A = \{a_1, a_2, ...\}$ is the set of all attributes associated with ct. Each attribute $a_i \in A$ is a triple a_i *={name,type,mode}* where name is the attribute name, type is the valid datatype for attribute values, and mode $\in \{o, m\}$ indicates whether specifying this attribute value at instantiation is optional or mandatory. An example of a credential type specification is as follows:

1. (Patrolman, <patrol_area, geospatial, m>, <badge_number, integer, m>)

2. (Police <precinct, string, m>, <rank, string, m>)

We use these credential type templates to instantiate new users' credentials. Values for mandatory attributes for a credential type as well as mandatory attributes for that credential type's parent types in the credential type hierarchy must be specified for each instance of that credential type. Instances of a specific credential type (user credentials) are created as follows:

1. *Bob = (Police, {(precinct,"183rd"),(rank, "sergeant") })*

2. *Jane= (Patrolman{(patrol_area ,"Richardson, TX"), (badge_number,20005665), (precinct,"183rd"),(rank, "lieutenant"})*

Credential expressions

Credential expressions are queries that allow a particular set of users to be specified based on their credential types and credential attribute values. In other words, credential expressions provide a means of specifying a subject not explicitly by subject id, but rather by the credential types associated with the subject and the values attached to these credential types for a particular instantiation of a subject. Given a set of credential types $CT = \{ct_1, ct_2, ... \}$ and the set of credential variables $CV = \{A(ct_1) \cup A(ct_2) \cup ...\}$ where $A(ct_i)$ is the set of attributes associated with each ct_i . $ct_i(x)$ is then considered a credential expression if $ct_i \in CT$ and $x \in SV$ where *SV* is the set of credential variables ranging over all subjects identifiers. $s_i(x)$ is also a credential expression describing the particular subject identified by s_i . Credential expressions may also be compound combinations of these simpler forms of credential expressions and literals joined by logical, geospatial, or temporal

operators (defined above). Specialized operators such as temporal and geospatial must be matched to their appropriate terms.

The following are examples of credential expressions:

1. Building_manager(x) - this expression requires the authorization subject to hold the building manager credential to access the data. Note that the values of the credential attributes are not taken into consideration in this example.

2. Peter(x) - this expression requires the subject to be identified as Peter to gain access

3. (Building_manager(x) $\wedge$ (name="Mr. Jones") $\wedge$ (office_number="23") – This expression requires the subject to hold the building manager credential as well has possess specific values for the credential attributes.

(Security_Officer(x) $\vee$ (patrol_area *contain* "building 1") – this is an example of use of a geospatial operator within a credential expression

Privilege Modes

Privilege modes dictate the type of access to be granted to the surveillance video object (table 1). The complex nature of video surveillance objects requires definition of a variety of modes which go beyond read and write access. As in the geospatial access control model we have subdivided the privilege modes into three categories browsing, copying and maintenance. Unlike text data in which browsing and copying amount to the same operation, allowing someone to view video data does not necessarily imply that this entity is allowed to copy or download this data. Furthermore, our proposed surveillance video objects are composed of several components including raw video, video annotation and the combination of the two, annotated video data. We must define privilege modes to access each of these. Additionally as with traditional access control mechanism we must be aware of release and aggregate inference and, therefore, define responsible access to combinations of data. Maintenance issues such as composition, modification and deletion of objects in the surveillance video database must also be taken into account as well as data manipulation techniques such as zoom should. We have provided for each of these various scenarios with the various privilege modes.

Because many privilege modes subsume the rights of other privilege modes we can model these with ordered hierarchical relationships denoted by $\prec_p$ such that $p_1 \prec_p p_2$ indicates that privilege mode p_1 is subsumed by privilege mode p_2. Maintenance privileges generally subsume copying privileges which generally subsume browsing privileges.

A partial ordering of the privilege mode rights is given by the following expression:

$$View \prec_p Zoom-in \prec_p Download \prec_p Update \prec_p Insert \prec_p Delete$$

Table 1: Table of Privilege Modes

Type	Privilege	Description
Browsing	View-annotation	Display Video annotation
	View-thumbnail	Display Video thumbnail
	View	Display Video
	Zoom-in	Display Zoomed-in video
Copying	Download	Download source object
	Download-annotation	Download annotation
Maintenance	Update	Modify object or object annotation
	Insert	Insert New Objects into Database
	Delete	Delete Surveillance video Objects

4.6 Authorizations

Authorizations are what allow us to specify our access control policy for the objects in our video surveillance database. An authorization is a 4 value tuple of the form <ce,ve,pr,t> where ce is the credential expressions specifying the subject credentials required to view the authorization object, ve is the video surveillance object expression specifying the authorization object(s) being considered, pr is the privilege modes or operations allowed on the data, and t is a temporal term representing the time period during which this authorization is valid.

Derived Authorizations

The properties of the hierarchical taxonomies used in defining surveillance video object types, semantic event types and semantic object types can be used to obtain implicit authorizations from the explicit authorizations specified as a part of the access control policy base. Additionally the relationships between the various privilege modes allow further extrapolation of authorizations.

For example, a policy authorization granting copying privileges to all objects containing events of type "rally" with high powered zoom to all users of credential type "Security Officer" allows derivation of numerous additional authorizations. We can generalize this policy to apply to video objects containing events more specific than "rally" in the event hierarchy, for instance "protest". Using the same kind of extrapolation, we can grant access to credential types that are more specific than the explicit authorization subject in the credential type hierarchy. An example of this would be the role of "Police Officer" which is a more specific concept than "Security Officer". The copying privilege implies that we should also grant read and browse privileges as due to the relationship between these privilege modes. Furthermore, high-powered zoom configurations imply that less detailed views of the scene should also be available to the authorization subject.

When used in combination these derivation rules can form a large and robust authorization base comprised of relatively few policy authorizations explicitly stated by the system user.

Formally, we can divide our derivation rules into three separate categories: privilege order derivation, hierarchy based derivation, and geographic area based derivation.

Privilege order derivation uses the partial privilege hierarchy to derive implied authorized privilege modes from explicitly stated ones. Given an authorization *a=<ce,ve,pr,t>* and a privilege mode *x* such that $x \prec_p pr$ then the derived authorization $a' =< ce, ve, x, t >$ holds for any *x*.

Hierarchy based derivation uses the surveillance video object hierarchy, semantic video object type hierarchy and semantic video event type hierarchy to derive authorizations.

Surveillance video object type hierarchy derivation:

Given an authorization *a=<ce,ve,pr,t>* and a object type *ot* such that $ot \prec_{OT} type(ve)$ then the derived authorization $a' =< ce, ve', pr, t >$ where $ve' = ve$ such that $type(ve') = ot$ holds for any such *ot*.

Semantic video object type hierarchy derivation:

Given an authorization a=<ce,ve,pr,t> and a semantic object type sot such that where then the derived authorization where such that holds for any such sot.

Semantic video event type hierarchy derivation:

Given an authorization *a=<ce,ve,pr,t>* and a semantic event type *set* such that $set \prec_{SET} x$ where $x \in events(ve)$ then the derived authorization $a' =< ce, ve', pr, t >$ where $ve' = ve$ such that $set \in events(ve')$ holds for any such *set*.

Geographic area based derivation allows objects whose geographic area is at the intersection of two explicitly authorized areas to be accessed with the same privilege mode as the greater of the two. Formally, Given two authorizations $a_i =< ce_i, ve_i, pr_i, t_i >$ and $a_j =< ce_j, ve_j, pr_j, t_j >$ such that $rectangle(ve_i) \cap rectangle(ve_j) \neq \varnothing$, $pr_j \prec_p pr_i$, $ce_i = ce_j$, and $t_i = t_j$ then the derived authorization $a' =< ce, ve', pr', t >$ where $ve' = ve_i \cap ve_j$ and $pr' = pr_i$ holds

4.7 Access Control Algorithm

User requests for surveillance video objects must be compared to the policy base of object authorizations before access can be granted. Furthermore, if the user request is not for a specific object but rather a query for a particular set of objects the system must be able to successfully reconcile the query criteria with the objects existing in the database. If the user request is authorized for some part (but not all) of the surveillance video object instead of denying the access entirely it is possible to post-process the data after retrieval and release only authorized portions to the user. Hence our access control process has three major components: Authorization, retrieval, post-processing and delivery.

The authorization component of the access control process considers user requests of the form $ur =< u, re, pr >$ where *u* is the user identifier, *re* is the object request in the form of a video object expression, and *pr* is the requested privilege mode for operation on the object. Given such a user request along with our authorization base *SVAB=* $\{a_1, a_2, ...\}$ where each authorization $a_i =< ce, ve, pr, t >$ our system should identify authorizations relevant to the requested object and determine whether the requesting subject has the appropriate credentials to satisfy the requirements of these authorizations. The first step of this process is surveillance video object evaluation. We compare the requested video object expression with each of the authorizations' video object expression to determine they pertain to overlapping sets of surveillance video objects. We next perform subject credential and privilege evaluation, by comparing the credentials of requesting user *u* with the credential expressions from the relevant authorization tuples. During this stage we also compare the requested privilege mode with the privilege mode in the authorization. If these criteria in the authorization are met we can retrieve, post-process and deliver the video data. Post-processing is done to enable access to only authorized portions of the video data. This is required especially if authorization policy specifies different levels of access based on semantic content of the video (i.e. some particular events or objects maybe classified for particular users).

4.8 Mapping To Semantics To Video Data

In order for the access control model proposed in this chapter to be as effective as possible the video object description must be as robust as possible (high availability of metadata). More specific and powerful queries and access control authorization can be written when the semantic content of the video is very well defined. The question still remains, how to map the low-level features of video into these high-level concepts such as events and objects? In other words, how do we bridge the semantic gap?

This issue with regards to event detection in video is addressed earlier in this thesis. Using techniques such as ours for behavioral analysis of video data in combination with developments in the area the area of object and face recognition will allow a robust semantic representation of video data that corresponds to the model described in this chapter. Such a representation will allow not only efficient access control in terms of human understandable concepts but also an improvement in the indexing and searching of these kinds of objects.

5. PRIVACY PRESERVING SURVEILLANCE

While our overall objective is to detect suspicious events, in section 4 we discussed how access could be controlled to sensitive objects. In this section we will discuss our approach to detect patterns but at the same time ensure confidentiality and privacy. That is while extracting pattern we want to ensure that confidential and private patterns do not go into the wrong individuals. The next step will be to integrate privacy preserving surveillance approach with the access control model described in section 4.

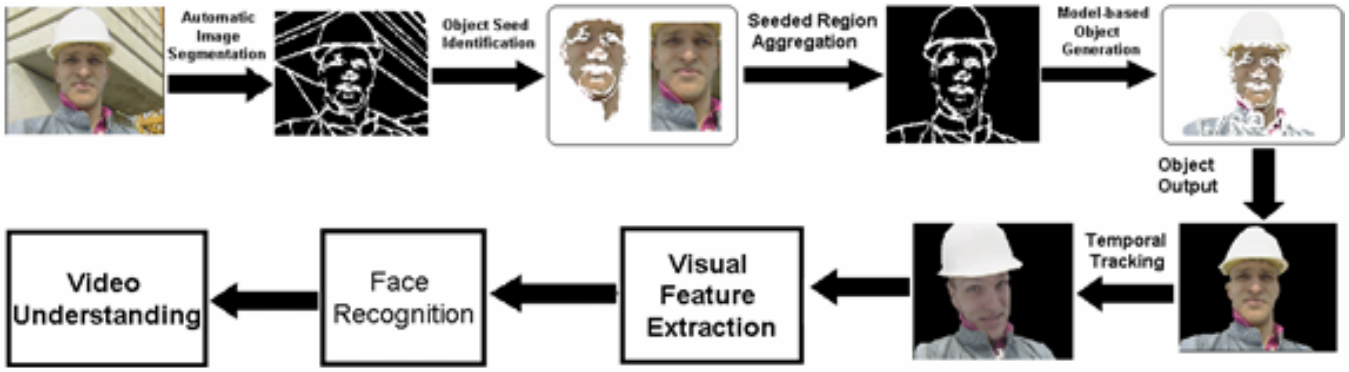

Figure 6: The flowchart of the proposed system for human object detection and tracking

The privacy or the confidentiality of a video largely depends on three inter-related factors: *video content sensitivity*, *potential receivers of the video*, and *usage of video contents by the potential receivers*. Obviously, all these three inter-related factors also depend from the confidentiality and privacy concerns of the party owning the video because confidentiality and privacy often have different meanings for different people or organizations or in different contexts. Each party must thus be able to define a vocabulary of privacy-sensitive video objects and must have available tools that, based on such vocabulary, are able to automatically identity the occurrence of such privacy-sensitive video objects before their release to the other parties.

Under our approach, the basic vocabulary of privacy-sensitive video objects will be predefined by the video owners that need to share and integrate their surveillance videos, To detect these privacy-sensitive video objects associated with each video shot, we will design a set of *automatic detection functions*; each of which able to detect only one certain type of privacy-sensitive video object from the basic vocabulary. Our detection function will take the following steps: (a) Use of the mean shift technique to achieve an automatic segmentation of video frames [9]; (b) Image region classification by using Support Vector Machines (SVM) classifiers to categorize the homogeneous image regions into two classes, that is, object regions *versus* non-object regions; (c) Model-based region aggregation for video object generation [9]; (d) Temporal object tracking to obtain the temporal relationships of object regions among video frames.

We use our detection function of the privacy-sensitive video object “human being” as an example to show how we can design our detection functions. Image regions with homogeneous color or texture will be first obtained by using the mean shift technique [9]. The homogeneous image regions that are related to the seed of the video object “human being” (i.e., human face), will be selected and labeled as the training samples for face detection. We will use *one-against-all* rule to label the training samples $\square_{gj}=\{X_l,\ L_j(X_l) \mid l=1,....,N\}$: positive samples for “human face” and negative samples.

The SVM classifier will be learned from these available training samples by maximizing the margin between the positive samples and the negative samples [7]. The learned SVM classifier will be used to classify the homogeneous image regions into: face regions versus non-face regions. Thus the connected face regions will be merged and aggregated as the corresponding object seed "human face". After the human face is detected, it will be treated as the seed of the video object "human being" and a model-based approach will be used to merge the related image regions for generating the video object "human being" [9]. To track the video object "human being" among the video frames, region-based motion estimation will be used to determine the temporal relationships of the object regions among the video frames. After the privacy-sensitive video objects are extracted, the original video streams will be decomposed into a set of component video objects such as human beings with race and gender, backgrounds, sensitive items, computer monitors, documents, and areas of interest.

To achieve a good balance between the benefits for surveillance tasks and the risks of security breaches, we propose a framework enabling multi-view privacy-preserving video sharing and integration. The main novelty of such framework is that different types of viewing privileges can be specified for different video contents to control the types of information released from videos. By combining those different viewing privileges with existing access control paradigms including new models such as the one discussed in section 4, articulated access control policies can be specified [8]. Figure 6 illustrates our architecture for privacy preserving surveillance. We need to integrate this system with the system proposed in Figure 1.

6. SUMMARY AND DIRECTIONS

In this paper we have addressed confidentiality and privacy for video surveillance databases. First we discussed an access control model and algorithms for confidentiality. Next we discussed privacy preserving video surveillance. As we have stated, we need t9o integrate the two approaches so that we can detect suspicious events as well as ensure privacy.

Our future work will proceed in two directions. One is to continue with the devel0pnebt if our access control model and compare it with models such as UCON and RBAC [15, 14]. Our goal is to extract the useful features from the multiple models so that we can develop an approach for surveillance databases. Second we will continue to enhance our research in privacy preserving surveillance. The idea here is to extract as many suspicious events as possible but at the same time maintain the privacy of individuals. We would also like to extend our approach to distributed surveillance databases.

7. REFERENCES

[1] ACLU, ``What's wrong with public video surveillance?'', 2002.

[2] R. Armitage, ``To CCTV or not to CCTV: A review of current research into the effectiveness of CCTV systems in reducing crime'', Technical Report, NACRO, London, 2002.

[3] V. Atluri, An authorization model for Geospatial databases, IEEE Transactions on Dependable and Secure Computing, 2004.

[4] Elisa Bertino, Ahmed K. Elmagarmid, Mohand-Said Hacid: A Logical Approach to Quality of Service Specification in Video Databases. Multimedia Tools Appl. 23(2): 75-101 (2004).

[5] Elisa Bertino, Jianping Fan, Elena Ferrari, Mohand-Said Hacid, Ahmed K. Elmagarmid, Xingquan Zhu: A hierarchical access control model for video database systems. ACM Trans. Inf. Syst. 21(2): 155-191 (2003).

[6] R. Collins, A. Lipton, H. Fujiyoshi, T. Kanade, ``Algorithm for cooperative multisensor surveillance'', Proc. IEEE, vol.89, no.10, 2001.

[7] J. Fan, Y. Gao, H. Luo, G. Xu, ``Salient objects: Semantic building blocks for image concept interpretation'', Int. Conf. on Image and Video Retrieval (CIVR'04), Dublin, Ireland, July 21-23, 2004.

[8] J. Fan, H. Luo, E. Bertino, ``Privacy-preserving video sharing for knowledge discovery", CIKM'05, Berlin, 2005.

[9] J. Fan, D.K.Y. Yau, A.K. Elmagarmid and W.G. Aref, ``Image segmentation by integrating color edge detection and seeded region growing'', IEEE Trans. on Image Processing, vol.10, pp.1454-1466,2001.

[10] B. Gelbord and G. Roelofsen, ``New surveillance techniques raise privacy concerns'', Comm. of ACM, vol.45, no.11, 2002.

[11] G. Lavee, Video Event Detection and Access Control, MS Thesis, University of Texas at Dallas, December 2005.

[12] G. Lavee, B. Thuraisingham, Suspicious Event Detection, ACM SIGKDD Multimedia Workshop, Chicago, IL, August 2005.

[13] G. Lavee, B. Thuraisingham, Event Detection in Surveillance Databases, Multimedia Tools, To appear, 2006

[14] Jaehong Park, Ravi Sandhu., "The UCONABC usage control model", ACM Transactions on Information and System Security (TISSEC), Volume 7, No 1, February 2004.

[15] R.S.Sandhu, E.J.Coyne, H.L.Feinstein and C.E. Youman, "Role-Based Access Control Models", IEEE Computer, Volume: 29, Issue: 2, Feb 1996.

[16] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y. Tian, A. Ekin, ``Blinkering surveillance: enabling video privacy through computer vision'', IBM TR W0308-109, 2003.

[17] B. Thuraisingham, Security for Multimedia Systems, Proceedings IFIP Database Security Conference, August 1990, Halifax, UK.