MLS-SMIL for Electronic Surveillance of Facilities with Multi-Level Security Requirements

[2]Naren Kodali, [3,4]Csilla Farkas and [1,2]Duminda Wijesekera
{nkodali|dwijesek}@gmu.edu, farkas@cse.sc.edu
[1]Center for Secure Information Systems, [2]Department of Information and Software Engineering,
George Mason University, MS 4A4, Fairfax, VA 22030.
[3]Information Security Laboratory, [4]Department of Computer Science and Engineering,
University of South Carolina, Columbia, SC 29208.

Abstract

We propose a framework that provides controlled distribution of live multimedia data to display devices with different access permissions for physical surveillance of multi-level secure facilities. Our proposal uses cameras and microphones as input devices and handheld radio linked displays as output devices. The geographical location of an input device determines its security level and the classification of the holder determines the security level of an output device. Our objective is to respect the security classifications whereby only those guards with sufficiently security classifications receive live media streams, but during emergencies, can receive pre-computed emergency instructions and/or cover stories. That facilitates avoiding disorder and obtaining appropriate support from individuals with lower levels of clearances. We propose using SMIL (XML like) formatted, multimedia feeds including cover stories, and show how to compose necessary multimedia document, compute views for each security classification, enforce access control and deliver to handheld devices respecting both wealthy run time semantics of multimedia as well as MLS security.

Key Words: Physical surveillance, SMIL, Secure multimedia, XML Security, Multi level security

1. INTRODUCTION

Electronic surveillance instruments [15,18,19] have been widely used to provide monitoring services. In some cases, monitored areas are accessible only to predefined groups of people, like command and control centers and missile storage facilities of a military base or traffic controller units of airports. It naturally follows that disclosure of live surveillance records must follow the access restrictions of the physical facilities. For example, consider an airport location. Passengers and employees can enter common areas, like ticketing, transportation facilities, and waiting areas. However, secured areas, like luggage transport and service stations, are available for airport employees only. The highest security areas, such as the air traffic control room, are accessible to specialized personnel only. We take the stance that people who are not authorized to access a physical location should not be able to view the

surveillance data of that location. However, during emergency operations, controlled dissemination of sensitive data may become necessary in order to obtain support services or to prevent panic. It has been shown that during crisis people require clear instructions so that their maximum cooperation is obtained. However, these instructions should not release unauthorized information or reveal the existence of such information. Therefore, it is necessary to develop methods and tools to allow selective access to surveillance feeds during normal and emergency operations. This paper proposes a framework to do so using appropriately secured SMIL formatted multimedia compositions.

Our proposal is to integrate monitoring and communication in a secure surveillance system that enforces access control restrictions on data while provides maximum availability. Our main contribution is the development of methodology to express multimedia compositions with their rich runtime semantics, techniques to enforce access control, and exploitation of cover stories to disseminate relevant material to unauthorized users during emergencies. For simplicity, we assume a multilevel security classification of physical areas and their corresponding surveillance data. Similarly, people accessing these facilities have security clearances. Employees and visitors are allowed to enter or view the surveillance feeds of a particular location (e.g., via broadcasts) only if they have the appropriate security clearance. We enforce that requirement during normal operations for guarding personnel. The main difference between a traditional MLS system and MLS for live surveillance feeds during day-today operations is the need to disseminate classified information to appropriate personnel for the latter. We propose that our multimedia surveillance system be equipped with a semantically rich, pre-orchestrated multimedia cover story repository, so that in emergencies cover stories can be released to lower security levels. For example, in case of fire in the top-secret area, personnel with lower clearances may be required to help, but the type of material that is burning should not be released to them. Also, public area visitors may be asked to leave the facilities in an ordered manner. In addition to enforcing MLS, we propose to record all sensory inputs obtained using the input devices, to be used for forensic analysis, as well as to improve the quality of cover stories.

Our model provides an integrated solution to manage surveillance devices, and to assert and collect audiovisual evidence for forensic analysis. We use XML-based multimedia composition framework, commonly referred to as SMIL, adapted for multi-level physical surveillance. The reason for selecting XML is guided by recent industrial trends. First, many browsers and off-the-shelf display and communication devices are becoming XML compliant [18,20]. Second, as mobile communication, using XML, becomes faster and more integrated, mobile devices are becoming available [18,19,20]. Third, toolkit support is becoming available to integrate XML compliant services [16,18]. Therefore, with the

right technological framework, our solution becomes portable to a wide range of general-purpose mobile multimedia devices such as those available in automobile navigation systems and hand-held devices. Multimedia communities have their own XML-like language known as the Synchronized Multimedia Integration language (SMIL)[3]. We extend upon existing proposals [5,14] to secure multimedia documents that are suitable for MLS secure surveillance. We show how normal and emergency operations of a MLS secure facility can be composed as a SMIL document enriched with proposed extensions. We take such a composition and construct views appropriate for different security classes, referred to as a MLS normal form of a SMIL document with appropriate security decorations. Then, given the runtime delays of an operational platform, we show how to generate an executable appropriate for that that runtime, which we call a display normal form of a SMIL document. We then encrypt media streams and transmitted them to intended recipients under normal and emergency operating conditions.

The rest of the paper is organized as follows Section 2 explains the application requirements of a general problem domain. In Section 3, we give a brief overview of related work. Section 4 describes SMIL, the XML-like language proposed for multimedia. In Section 5, we present our framework and compile time issues and runtime activities including encryption and resource management are explained in Section6. Section 7 concludes the paper and recommends future improvements.

2. RUNNING EXAMPLE

Figure 1 shows a hypothetical research facility with varying levels of sensitivity. Assume that the area enclosed by the innermost rectangle ABCD contains weapons with highest degree of sensitivity and is accessible (and therefore guarded by) personnel with the highest level of clearance, say top secret (TS). The area between the rectangles PQRS and ABCD is classified at medium level of sensitivity and therefore requires personnel with secret (S) security clearances. The area external to PQRS contains least sensitive material, and can be accessed by unclassified personnel, like visitors and reporters. We classify the areas into *Top-Secret (TS)*, *Secret (S)* and *Unclassified (UC)* security levels with application domains, e.g., *Dom* as categories. Security labels form a lattice structure. For simplicity, we omit the application domain and use *TS*, *S*, and *UC* as security labels. The area inside ABCD is TS, the area inside of PQRS, but outside of ABCD is S, and the area outside PQRS is UC. Employees, guards, support services personnel, and general public have $TS \geq S \geq UC$ clearances, where $\geq$ corresponds to the dominance relation defined in MLS systems. As depicted in Figure 1, an area with higher level of sensitivity is a sub-part of areas with all lower levels of sensitivities. Therefore, a guard with top-secret

clearance may be used in the classified area, but not vice versa. For electronic surveillance purposes, cameras (infrared and normal light) and other devices such as microphones are situated throughout the facility. Multimedia streams emanating from these devices are continuously used to monitor the facility. We propose a design where all multimedia data is transmitted to a centralized control facility and then directed to handheld devices of appropriate security personnel.

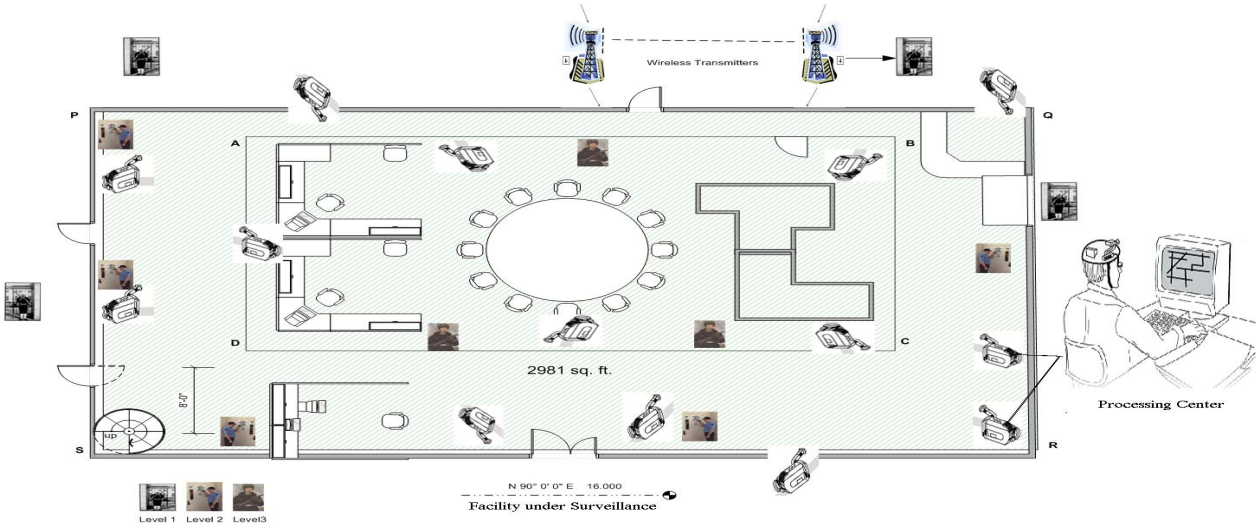

Figure 1: An Example MLS-Secure Facility

3. RELATED WORK

A distributed architecture for multi-participant and interactive multimedia that enables multiple users to share media streams within a networked environment is presented in [1]. In this architecture, multimedia streams originating from multiple sources can be combined to provide media clips that accommodate look-around capabilities.

Multilevel security (MLS) has been widely studied to ensure data confidentiality, integrity, and availability [4]. MLS systems provide controlled information flow based on the security classification of the protection objects (e.g., data items) and subjects of the MLS system (e.g., applications running in behalf of a user). To provide information confidentiality, data is allowed to flow only from low security levels to higher security levels [22]. Although our approach to provide controlled information flow in real-time multimedia systems is based in concepts similar to MLS, the developed methods and techniques are also applicable in other security models, like Role-Based or Discretionary Access Control models [23].

Regulating access to XML formatted text documents has been actively researched in the past few years offering a multitude of solutions. Bertino et al. [7,8,9], have developed Author-X, a Java based system to secure XML documents that enforces access control policies at various granularities and corresponding user credentials. Author-X encodes security policies for a set of XML documents in an XML file referred to as the policy base. They permit both permissions and prohibitions. Damiani et al. [11,12,13] developed an access control model where the tree structure of XML documents is exploited using XPATH expressions to control access at different levels of granularity. The smallest protection granularity is an XPATH node, and security policies specify permissions or prohibitions to all children objects of a node. Fine-grained accesses are specified using XML markup where a subject is a user who is generally a member of one or more user groups, and an object is any node defined by XPATH. Gabillon et al. [6] suggest an alternative to Damiani et al [11,13], where authorization rules related to a specific XML document are first defined on a separate style sheet, and then translated to an (eXtensible Stylesheet Language) XSL sheet.

Damiani et al. [12, 14] also discuss feature protection of XML format images. Its primary focus is controlled dissemination of sensitive data within an image. In [14] they propose an access control model with complex filtering conditions. This model uses SVG to render the map of a physical facility. While this model could be used to represent our model, it has limitations when compared to flexibility and adaptability to certain issues particular to physical security in the multilevel hierarchy. Bertino at al. [5] provides a security framework to model access control in video databases. They provide security granularity, where objects are sequences of frames or particular objects within frames. The access control model is based on he concepts of security objects, subjects, and the permitted access modes, like viewing and editing. The proposed model is provides a general framework of the problem domain, however it is not explained how access control objects to be released are formalized and enforced. Stoica at al. [4] present the concept of cover stories in the XML context. Their aim is to hide the existence of non-permitted data from the naïve user. The motivation of the work is the need to provide secure release of multilevel XML documents and corresponding DTD files where security sensitivity is not monotonically increasing along all paths originating from the node. The authors provide techniques how to modify an MLS/XML document to release non-sensitive data in a manner that is semantically correct and inference free. While their model addresses the need of multimedia, their approach does not incorporate semantics of multimedia.

None of the above approaches are completely satisfactory for surveillance multimedia. They primarily address textual documents and exploit the granular structure of XML documents. Multimedia for various reasons as discussed above has to be treated differently because there is a sense of temporal synchrony and continuity involved. Synchronization and integration of different and diverse events to produce sensible information is non-trivial when compared to textual data. The process of retrieval without losing the sense of continuity and synchronization needs sophisticated techniques and algorithms which all of the above models do not completely address.

Substantial amounts of contemporary research addresses real-time moving object detection and tracking from stationary and moving camera platforms, object pose estimation with respect to a geospatial site model, human gait analysis, recognition of simple multi-agent activities, real-time data dissemination, data logging and dynamic scene visualization [20,21]. While they have offered valuable directions to our research model, they do not provide a comprehensive solution to physical security.

4. THE SYNCHRONIZED MULTIMEDIA INTEGRATION LANGUAGE (SMIL)

SMIL [3] is an extension to XML developed by W3C to author presentations, allowing multimedia components such as audio, video, text and images to be integrated and synchronized to form a presentation. The distinguishing features of SMIL over XML are the syntactic constructs for timing and synchronization streams with qualitative requirements. In addition, SMIL provides a syntax for spatial layout including constructs for non-textual and non-image media and hyperlink support. In this section, we explain those SMIL constructs that are relevant for our application, and show how they can be used to specify a multimedia document satisfying the application needs stated in Section 3.

SMIL constructs for synchronizing media are <seq>, <excl> and <par>. They are used to hierarchically specify synchronized multimedia compositions. The <seq> element plays the child elements one after another in the specified sequential order. <excl> specifies that its children are played one child at a time, but does not impose any order. The <par> element plays all children elements as a group, allowing *parallel* play out. For example, the SMIL specification *<par><video src=camera1><audio src = microphone1></par>* specify that media sources camera1 and microphone1 are played in parallel.

In SMIL the time period that a media clip is played out is referred to as its *active duration*. For parallel play to be meaningful, both sources must have equal active durations. When clips do not have same active durations, SMIL provides many constructs to make them equal. Some examples are *begin* (allows to begin components after a given amount of time), *dur* (controls the duration), *end* (specifies the ending

time of the component with respect to the whole construct), *repeatCount* (allows a media clip to be repeated a maximum number of times). In addition, attributes such as *syncTolerance* and *syncMaster* controls runtime synchronization, where the former specifies the tolerable mis-synchronization (such as tolerable lip-synchronization delays) and the latter specifies a master-slave relationship between synchronized streams. In this paper, we consider only the basic forms of synchronization construct (that means, we do not specify syncMaster and syncTolerance. Thus we assume that components of <par> have equal play out times and they begin and end at the same time.

An important construct that we use is <*switch*> allowing one to switch among many alternatives compositions listed among its components. These alternatives are chosen based on the values taken by some specified attributes. For example, *<switch> <audio src="stereo.wav" systemBitrate≥25><audio src="mono.wav" systemBitrate ≤ 25></switch>* plays stero.wav when the SMIL defined attribute *systemBitrate* is at least 25 and *mono.wav* otherwise. We use this construct to specify our surveillance application. In order to do so, we define two custom attributes *customTestMode* that can take values "normal" and "emergency" and *customTestSecurity* that take any value from ("TS","S","UC"). The first attribute is used to indicate the operating mode that can be either normal or emergency and the second attribute indicates the security level of streams that can be top secret, secret or unclassified. SMIL also requires that every application-defined attribute (custom attribute in SMIL terminology) have a title and a default value. It further has a special flag *override* that takes the value *hidden* or *visible*. When override takes the value hidden, the player is not allowed to change the value of the custom attributes. That feature is useful in specifying security attributes that are not to be altered by SMIL players.

Figure 2 shows a simplified example of a SMIL specification for surveillance. As the figure shows, the file consists of two sections, where the first section defines the custom attribute *customTestMode* with values "Normal" and "Emergency". Because the second and the fourth lines of Figure 2 specifies that customTestMode is *hidden*, the value of this attribute corresponding to each stream cannot be reset later. The second part of the file consists of a switch statement consisting of collection of media streams connected by <par> constructs. Notice that inside the <switch> statement consists of two sections where the first one begins with the line <par customTestMODE= "Normal"> and the second one begins with the line <par customTestMODE= "Emeregency">. That specifies that the streams inside be shown under normal and emergency operating conditions.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<customAttributesMODE>
        <customTestMode="Normal" title="Normal Mode"
          defaultState="true" override="hidden"
         <customTestMode id="Emergency" title="Emergency Mode"
          defaultState="true" override="hidden"
<customAttributesMODE>
<customAttributesSecurity>
        <customTestSecurity id="TS" title="Top-Secret"
          defaultState="true" override="hidden"/>
        <customTestSecurity id="S" title="Secret"
          defaultState="true" override="hidden"/>
        <customTestSecurity id="UC"  title="Unclassfied"
          defaultState="true" override="hidden"/>
</customAttributesSecurity>
<body>
<switch>
//Classification is TS(Top-Secret)
<par customTestMODE= "Normal">
<video src="CameraTS1.rm" channel="video1" customTestSecurity="TS"/>
<audio src="CameraTS1.wav" customTestSecurity="TS" />
//Classification is S(Secret)
<video src="CameraS1.rm" channel="video1" customTestSecurity="S"/>
<audio src="CameraS2.wav" customTestSecurity="S"/>
//Classification is U(Unclsssified)
<video src="CameraU1.rm" channel="video2" customTestSecurity="S"/>
<audio src="CameraU1.wav" customTestSecurity="S" />
</par>
<par customTestMODE= "Emergency">
//All 3 above together (Total of 6 feeds)
//Here are the secret cover stories
<par>
<video src="CoverstoryTS-to-S1.rm" channel="video1" id="TS-to-Secret"
customTestSecurity="S"/>
<audio src="CoverstoryTS-toS1.wav" customTestSecurity="S"/>
</par>
//Here are the unclassified cover stories
<par>
<video src="CoverstoryTS-to-U1.rm" channel="video1" id="TS-toUC1"
customTestSecurity="U"/>
<audio src="CoverstoryTS-to-U1.wav" customTestSecurity="U"/>
<video src="CoverstoryS-to-U1.rm" channel="video1" id="Secret-toUC1"
customTestSecurity="U"/>
<audio src="CoverstoryS-to-U1.wav" customTestSecurity="U"/>
</par>
//Followed by normal the TWO UC camera feeds.
</switch>
</body>
</smil>
```

Figure 2: SMIL Specification for Figure 1

In this example, each area has a camera and a microphone to record audio and video streams to be transmitted to appropriate guards. They are named CameraTS1.rm, CamerU1.wav etc. The security classification of each source is identified by the application defined SMIL attribute *customTestSecurity*. For example, <video src="CameraTS1.rm" channel="video1" customTestSecurity="TS"/> specifies that the video source named *CameraTS1.rm* has the Top Secret security level. The intent being that this source is to be shown only to top-secret guards.

As the second half of Figure 2 shows, there are three audio-visual cover stories named *CoverstoryTS-to-S1.rm* to *CoverstoryS-to-UC1.wav* are shown with the appropriate security level specified with the attribute customTestSecurity. The main composition is encoded using a <switch> statement that is to be switched based on the operating mode (normal or emergency).

5. PREPROCESSING SMIL DOCUMENTS TO ENFORCE MLS SECURITY MODELS

Surveillance requirements, such as the example given in Figure 2 specifies which multimedia sources have to be displayed under the two operating conditions. We assume that the source document specifies the security label of each source and that MLS policies are used to ensure that guards are permitted to view only those multimedia sources that are dominated by the guards' security clearances. For this, we preprocess a given MLS multimedia document and produce views that are permitted to be viewed by

guards for each security classification. Then, we separately encrypt and broadcast multimedia documents for each category, to the appropriate locations by efficient use of bandwidth.In order to achieve this objective, we first transform every SMIL document with proposed security and mode attributes to three SMIL documents, where all security labels in each document consists of solely one *customTestSecurity* attribute, namely the one that is appropriate to be seen by guards with the label value. We now formally state and prove that this can be done for an arbitrary SMIL document with our security labels.

Definition 1 (MLS Normal Form)

We say that a SMIL specification S is in Multi Level Secure Normal Form (MLSNF) if it is of one of the following forms:

1. It is of the form <par> Cts(S) Cs(S) Cu (S) Cud(S) Cod</par> where all attributeTestSecurity attributes in Cts(S), Cs(S), Cu(S) are respectively TS, S and U. In addition, Cud(S) has no attributeTestSecurity and Cod(S) has two different value set for attributeTestSecurity.
2. It is of the for <par> Cts(S) Cs(S) Cu (S) Cud(S) Cod(S)</par> with one or two components of <par> may be missing. Here Cts(S), Cs(S) and Cu(S), Cud(S) Cod(S) satisfy requirements stated above.
3. It is of the form Cts(S), Cs(S), Cu(S), Cud(S), Cod(S) where Cts(S), Cs(S), Cu(S), Cud(S) and Cod(S) satisfy requirements stated above.

We say that Cts(S), and Cs(S) and Cu(S) are respectively the top secret, secret and unclassified views of the specification S. Cud(S) is the view with missing security classifications and Cod(S) is the view with contradictory security classifications. □

As stated in Definition 1, a SMIL specification in MLSNF is one that is parallel composition of at most three specifications, where each specification belongs to one security class, that are said to be the views corresponding to the respective security classes. Notice that in Definition 1, the latter two cases are degenerate cases of case 1 where one or more views of the specification become null.

In attempting to create views from an arbitrary SMIL document, one encounters two undesirable situations. The first is the missing security classifications resulting in a non-null Cud(S). The other is the situation with contradictory security classification due to over specification. An example under specified SMIL specification is <audio src= “myAudio.wav”>, and an example contradictory specification is <video src= “myMovie.rm” attributeTestSecurity=TS attributeTestSecurity=S>. Thus, it is tempting to avoid such situations by applying completeness and conflict resolution policies [25] designed to be used in XML formatted and databases. Note, that completeness and conflict resolution polices were intended to be used for inheritance hierarchies. Because SMIL hierarchies are not due to inheritances and instead they are syntactic constructs for media synchronization, blindly applying such policies to resolve under

and over specification of SMIL documents destroys the synchronized play out semantics of media streams. Designing appropriate policies to address over and under specification SMIL constitute some of our ongoing work. In this paper, we use the neutral policy of discarding under and over specified fragments Cud(S) and Cod(S) of a SMIL specification S. We now give an algorithm that transforms a given algorithm into its MLSNF.

Algorithm 1 (toMLSNF)
Input: SMIL composition with security decorations
Output: SMIL document in MLSNF

1. If S is a media stream (such as <video …> or <audio …>) with possibly an attributeTestSecurity attribute. Then:
   a. If attributeTestSecurity=TS, then Cts(S)=S, Cs(S)=ϕ and Cu(S)=ϕ and Cu(S)=ϕ..
   b. If attributeTestSecurity=S, then Cts(S)=ϕ , Cs(S)=S, and Cu(S)=ϕ.
   c. If attributeTestSecurity=U, then Cts(S)=ϕ , Cs(S)= ϕ, and Cu(S)=S.
   d. If attributeTestSecurity does not exists in S, then Cts(S)=ϕ , Cs(S)=ϕ, Cu(S)=ϕ and Cud(S)=S Cod(S)=ϕ.
   e. If there is more than one instance of attributeTestSecurity in S then Cts(S)=ϕ , Cs(S)=ϕ, Cu(S)=ϕ and Cud(S)= ϕ Cod(S)=S.
2. If S is <seq>S1 S2</seq> then,
   a. Cts(S)=<seq>Cts(S1) Cts(S2) </seq>
   b. Cs(S)=<seq>Cs(S1) Cs(S2) </seq>
   c. Cu(S)=<seq>Cu(S1) Cu(S2) </seq>
   d. Cud(S)=<seq>Cud(S1) Cud(S2) </seq>
   e. Cod(S)=<seq>Cod(S1) Cod(S2) </seq>

However, if either of Cx(Si) are empty for some x ε{TS,S,U,UD,OD} and i ε{1,2}, then Cx(Si) in the right hand sides above must be substituted by NULL(Si) where NULL(Si) is defined as
<type src=empty.type, attributeTestSecurity=Y,dur=Z type> where Z and Y are respectively durations and the attributeTestSecurity attribute values appearing in Si.

3. If S is <par>S1 S2</par> then,
   a. Cts(S)=<par>Cts(S1) Cts(S2) </par>
   b. Cs(S)=<par>Cs(S1) Cs(S2) </par>
   c. Cu(S)=<par>Cu(S1) Cu(S2) </par>
   d. Cud(S)=<par>Cud(S1) Cud(S2) </par>
   e. Cod(S)=<par>Cod(S1) Cod(S2) </par>
4. If S is <switch>S1 S2</switch> then,
   a. Cts(S)=<seq>Cts(S1) Cts(S2) </seq>
   b. Cs(S)=<seq>Cs(S1) Cs(S2) </seq>
   c. Cu(S)=<seq>Cu(S1) Cu(S2) </seq>
   d. Cud(S)=<seq>Cud(S1) Cud(S2) </seq>
   e. Cod(S)=<seq>Cod(S1) Cod(S2) </seq>

Then let MLSNF(S) = <par>Cts(S) Cs(S) Cu(S) Cud(S) Cod(S)</par>.

Algorithm 1:toMLSNF (Conversion to a Normal Form)

We now have to ensure that Algorithm 1 preserves semantics. That is, top secret, secret and unclassified viewers of a specification S will view Cts(S), Cs(S) and Cu(S) respectively. This proof would be easy, provided that we have a formal operational semantics for SMIL. While providing such semantics is not difficult, it does not exist yet. Therefore, while we are working on it, we provide a rudimentary operational semantics for the purposes of showing that our algorithms work as expected.

5.1. A Rudimentary Operational Semantics for SMIL

In this section, we provide a simple operational semantics for media streams and SMIL documents constructed using <par>, <seq> and <switch> commands. The sole objective of this exercise is to show that Algorithm 1 transforms a SMIL document to a collection of ones that remain invariant with respect to this semantics. The latter is referred to as semantic equivalence [26]. Following customary practices in programming language semantics, our operational semantics and the proof of semantic equivalence

will be inductive in nature. It is worth noting that our semantics is only applicable to our application scenario and syntactic constructs, and its extension to other purposes and constructs form our ongoing work.

Definition 2 (Timed Display Instance)

We say that a quadruple (S, T-begin, T-end, Security Set) is a timed display instance provided that:

1. S is a basic media element with a finite active duration $D \geq 0$.
2. T-begin $\leq$ T-end are arithmetic expressions of a single real variable **t** satisfying T-end=T-begin + D.
3. Security set a subset of {TS,S,U} consisting of *attributeTestSecurity* attribute values of S.
4. We say that a set of timed display instances is a timed display set provided that there is at least one timed display element with t as its T-begin value.
5. Taken as expressions containing the variable t, the smallest T-begin value of a timed display set is said to be the *origin* of the timed display set. We use the notation O(TDI) for the origin of the timed display set TDI.
6. Taken as expressions containing the variable t, the largest T-begin value of a timed display set is said to be the *end* of the timed display set. We use the notation E(TDI) for the end of the timed display set TDI.

The following two elements tdi_1 and tdi_2 are examples of timed display instances.

1. tdi-1 = (<video, src= "myVideo.rm", dur=5, attributeTestSecurity=TS>, t, t+7, {TS})
2. tdi-2 = (<audio, src= "myAudio.rm", dur=10, attributeTestSecurity=U>, t+7, t+17, {U})

Therefore, {tdi-1,tdi-2} is timed display set with its origin t and end t+17. The intent here is to consider TDI={tdi-1,tdi-2} as a possible playout of the SMIL specification <seq><video, src= "myVideo.rm", dur=5, attributeTestSecurity=TS>, <audio, src= "myAudio.rm", dur=10, attributeTestSecurity=U> </seq> that begin at an arbitrary but thereafter fixed time t and ends at t+17.

Now we describe some algebraic operations on timed display sets that are necessary to complete the definition of our operational semantics of SMIL. The first is that of *origin substitution* defined as follows.

Definition 3 (Algebra of Timed Display Sets 1: Substitution)

Suppose TDS is a timed display set with the formal time variable t and s is any arithmetic expression possibly containing other real valued variables. Then TDS(s/t) is the notation for the timed display set obtained by syntactically substituting all timing values (that is T-begin and T-end values) of elements of TDI.

For the example TDI given prior to Definition 3, TDI(2t+7/t) consists of {tdi-1(2t+7/t),tdi-2(2t+7/t)} where tdi-1(2t+7/t) and tdi-2(2t+7/t) are defined as:

1. tdi-1(2t+7/t) = (<video, src= "myVideo.rm", dur=5, attributeTestSecurity=TS>, 2t+7, 2t+21, {TS})
2. tdi-2(2t+7/t) = (<audio, src= "myAudio.rm", dur=10, attributeTestSecurity=U>, 2t+21, 2t+31, {U})

The reason for having Definition 3 is that in order to provide formal semantics for the <seq> operator, it is necessary to shift the second child of the <seq> by the time duration of its first child and repeat this procedure for all of <seq>'s children. To exemplify the point, the first example the TDI={tdi-1,tdi-2} is infact {tdi-1} U TDI'(t+7/t) where TDI' is given by tdi' = (<audio, src = "myAudio.rm", dur=10, attributeTestSecurity = U>, t, t+10, {U}). We are now ready to obtain operational semantics for SMIL specifications, provide the following assumptions are valid.

5.1.1. Assumptions about SMIL constructs and the reasons for making them

1. <par> construct is applied to components with equal active durations.

   The reason for this assumption is that we are assuming that all components of <par> to have the same begin time and the same end time. We can relax this assumption and still obtain the same result, but the proofs will be a bit more complicated.
2. We do not use the <excl> construct.

   This construct introduces non-determinism. That makes our semantics inapplicable. Appropriate semantics for the <excl> construct can be obtained if each object is mapped to a set of possible interpretations. This is the same solution offered for the semantics of non-deterministic programs in denotational semantics of concurrency – namely the power domain construction [27].

Using these assumptions, we now proceed to provide formal semantics.

Definition 4 (Basis Mapping)

Suppose M is the set of basic media elements of S. Then any mapping [[ ]] from M to a set of Timed Display Instances TDI is said to be a basis mapping for a denotation iff all T-begin elements of M have the same value t, where t is a real variable. Then we say that [[ ]] is a basis mapping parameterized by t.

Lemma 1 (Existence of basis mappings)

Suppose M is a set of basic media streams with time durations. Then M has a basis mapping.

Proof:

For each media stream m= <type, src= "...", dur=value, attributeTestSecurity= "..." type>, in M, let [[M]] map to (m, t, t+value, {Att Values}). Then [[ ]] is a basis mapping

We now use a basis mapping to define operational semantics of any SMIL specification S as follows.

Definition 5 (Operational Semantics for SMIL)

Suppose S is a SMIL specification and [[ ]] is a basis mapping for the basic media elements B of S with the formal parameter t. Then we inductively extend [[ ]] to S as follows.

1. [[Null]] = Φ.
2. [[<seq> S1 S2</seq>]] = [[S1]] U [[S2]](end([[S1]])/t)
3. [[<par> S1 S2</par>]] = [[S1]] U [[S2].
4. [[<switch> S1 S2 </switch>]] = [[S1]] if S1 satisfies the attribute of the switch.
   = [[S2]] otherwise if S2 satisfies the attribute of the switch.
   = Φ otherwise.

We now say that the extended mapping [[ ]] is a semantic mapping parameterized by t.

It is our position that the informal definition given the SMIL specification is captured by our operational semantics, provided we are able to evaluate the *attribute of the switch*. This can be easily formalized using customary practices of program language semantics, and is therefore omitted here for brevity.

We now formally state and prove the semantic equivalence of Algorithm 1. That shows that rewritten specification has the same operational semantics as the original. That we offer as the correctness argument for the rewrite.

Theorem 1 (Correctness of Algorithm 1)

Suppose that S is a SMIL specification and [[ ]] is a semantic mapping parameterized by t. Then [[S]] = [[MLSNF(S)]].

Proof:

As stated earlier, this proof also proceeds by induction on the structure of S. Thus, for the sake of brevity, we show one base case and one inductive case.

1. An Example Base Case
Suppose S is <type src=" ", dur=n, attributeTestSecurity="S" type>. Then, by Algorithm 1,
Cts(S) = Null, Cs(S) = S, Cu(S) = Null, Cud(S) = Null and Cod(S) = Null. Therefore,
[[<par>Cts(S) Cs(S) Cu(S) Cud(S) Cod(S) </par>]]= [[<par>Null S Null Null Null</par>]] =
[[Null]] U [[S]] U [[Nul1]] U [[Nul1]] U [[Null]] = [[S]]. Hence [[MLSNF(S)]] = [[S]].

2. An Example Inductive Case
Suppose S is <seq> S1 S2</seq>. Then, from Algorithm 1,
[[MLSNF(S)]] =[[<par>Cts(S) Cs(S) Cu(S) Cud(S) Cod(S) </par>]]
= [[Cts(S)]] U [[Cs(S)]] U [[Cu(S)]] U [[Cud(S)]] U [[Cod(S)]]
= [[<seq>Cts(S1) Cts(S2)</seq>]] U [[<seq>Cs(S1) Cs(S2)</seq>]]
U [[<seq>Cu(S1) Cu(S2)</seq>]] U [[<seq>Cud(S1) Cud(S2)</seq>]]
U [[<seq>Cod(S1) Cod(S2)</seq>]]
= [[Cts(S1)]] U [[Cts(S2)]](end([[Cts(S1)]])/t) U[[Cs(S1)]] U [[Cs(S2)]](end([[Cs(S1)]])/t)
U [[Cu(S1)]] U [[Cu(S2)]](end([[Cu(S1)]])/t) U[[Cud(S1)]] U [[Cud(S2)]](end([[Cud(S1)]])/t)
U [[Cod(S1)]] U [[Cod(S2)]](end([[Cod(S1)]])/t)
Conversely,
[[S]] = [[<seq> S1 S2 </seq>]] = [[S1]] U [[S2]](end(S1)/t)
= [[Cts(S1)]] U [[Cs(S1)]] U[[Cu(S1)]] [[Cud(S1)]] U [[Cod(S1)]]
U([[Cts(S2)]] U [[Cs(S2)]] U[[Cu(S2)]] [[Cud(S2)]] U [[Cod(S2)]]) (end(S1)/t)
by the inductive assumption.
But notice that ([[Cts(S2)]] U [[Cs(S2)]] U[[Cu(S2)]] [[Cud(S2)]] U [[Cod(S2)]]) (end(S1)/t)
= [[Cts(S2)]](end([[Cts(S1)]])/t) U [[Cs(S2)]](end([[Cs(S1)]])/t) U [[Cu(S2)]](end([[Cu(S1)]])/t) U
[[Cud(S2)]](end([[Cud(S1)]])/t) U [[Cod(S1)]] U [[Cod(S2)]](end([[Cod(S1)]])/t)
Therefore [[MLSNF(S)]] = [[S]], thereby justifying the inductive case.

On rewriting the example in Figure 2 in the MLS Normal form we create at the different views for each of the following cases each represented as a separate SMIL document. In the Figure 2 below we have

the format of such a specification denoting the entire structure of a “Top-Secret” view in the normal mode and a “Secret” view in the emergency mode.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<customAttributesMODE>
        <customTestMode="Normal" title="Normal Mode"
          defaultState="true" override="hidden"
         <customTestMode id="Emergency" title="Emergency Mode"
          defaultState="true" override="hidden"
<customAttributesMODE>
<customAttributesSecurity>
        <customTestSecurity id="TS" title="Top-Secret"
          defaultState="true" override="hidden"/>
        <customTestSecurity id="S" title="Secret"
          defaultState="true" override="hidden"/>
        <customTestSecurity id="UC"  title="Unclassfied"
          defaultState="true" override="hidden"/>
</customAttributesSecurity
<seq>
<switch>
<par customTestMode =”Normal” customTestSecurity = “TS”>
<par>
<video src="Tsameral.rm" channel="video1" dur="45s"/>
<audio src="TSCameral.wav"  />
</par>
<par>
<video src="TSCamera2.rm" channel="video1" />
<audio src="TSCamera2.wav"/>
</par>
</par>
<par customTestMode =”Normal” customTestSecurity = “S”>
XXXX//Normal Form View for Normal Mode “S” Class
</par>
<par customTestMode =”Normal” customTestSecurity = “UC”>
XXXX//Normal Form View Normal Mode “UC” Class
</par>
<par>
<par customTestMode =”Emergency” customTestSecurity = “TS”>
 XXXX//Normal Form View for Normal Mode “TS” Class
</par>
<par customTestMode =”Emergency” customTestSecurity = “S”>
<video src="SCameral.rm" channel="video2" dur="25s"/>
<audio src="SCameral.wav"  />
</par>
<par>
<video src="Scamera2.rm" channel="video2"/>
<audio src="Scamera2.wav"  />
</par>
<par>
<video src="CoverstoryTS1.rm" channel="video1" id="TSCoverstory1"/>
<audio src="CoverstoryTS1.wav"  />
</par>
<par>
<video src="CoverstoryTS1.rm" channel="video1" id="TSCoverstory1"/>
<audio src="CoverstoryTS1.wav"  />
</par>
<par customTestMode =”Emergency” customTestSecurity = “UC”>
 XXXX//Normal Form View for Emergency Mode “UC” Class
</par>
</switch>
</seq>

Each   VIEW and will be made into a SMIL document and named as follows

ModeNClassTS.smil //SRM TS Normal    ModeEClassTS.smil    //SRM TS Emergency
ModeNClassS.smil //SRM S Normal    ModeEClassS.smil    //SRM S Emergency
ModeNClassUC.smil    //SRM UC Normal ModeEClassUC.smil // SRM UC Emergency
```

Figure 3:MLS Normal Form Specification of Figure 2.

6. THE RUNTIME BEHAVIOR OF MLS SURVEILLIANCE SYSTEM

In the most general case, a SMIL specification in MLSNF is of the form <par> Cts Cs Cu Cod Cud </par> where Cts Cs Cu Cod and Cud respectively have top secret, secret, unclassified, over specified and under specified security levels. How one resolves under specification and over specification is a matter of policy, and is not addressed in this paper. Independently, Cts, Cs, Cu are to be shown to guards with top secret, secret, and unclassified clearances. In addition, in order to respond to emergencies, these specifications have a mode switch encode using a custom attribute *attributeTestMode*. As observed in Figure 2, this attribute is to be evaluated at the beginning of a <switch> statement. That is unsatisfactory for intended purposes, after this switch statement is executed, the operating mode could vary many times. Because the <switch> is evaluated only once, the SMIL specification is now oblivious to such changes in application situations. In this section, we show how to rewrite a SMIL document with one <switch> statement for changing a mode to that one that makes the *attributeTestMode* be evaluated at regular intervals. Although in theory any system could switch its operating mode in an arbitrarily small time intervals, practical considerations limits this interval to a minimum. This minimum switching

granularity may depend upon many parameters such as hardware, software and the inherent delays in of switching on firefighting and other emergency related equipment. Therefore, given a switching delay D, we rewrite the given SMIL document so that the mode attribute *attributeTestMode* re-evaluated every D time units. How that is done is discussed in the next section.

6.1. Display Normal Form Informally

The following SMIL specification in Figure 4, has the same structure as the specification in Figure 2 and Figure3.If we want to break up this specification so that the *attributeTestMode* is tested each D units of time and the switch reevaluated, then the code can be translated as follows(right hand side of Figure4).

S_1 =<switch> <par attributeTestMode= "normal"> XX </par> <par attributeTestMode= "emergency"> </par> </switch>	**S_2** = <par dur=D, repeatCount="indefinite"> <switch> <par attributeTestMode="normal"> XX </par> <par attributeTestMode="emergency"> YY </par> </switch> </par>

Figure 4: Mode Evaluation Semantics.

Notice that the outer <par> construct specifies that enclosing specification be executed for duration of D time units and repeated indefinitely. However, the outer <par> construct has only one element, namely the switch. Therefore, the <switch> construct is executed for infinitely many times, and each time the attributeTestMode is tested. Given a SMIL specification with the *attributeTestMode* specified in the form where the switch is reevaluated every D time units is said to be in display normal for the attribute *attribbuteTestMode* and time duration D. We have now informally shown that every SMIL document where the *attribbuteTestMode* is used in the stated form can be translated into its display normal form.

We stress the informal nature of our argument because of our commitment to the specified operational semantics. A brief inspection will show the reader that our translation into display normal form is not semantically equivalent under semantics provide in definition 6. However this semantics can be enhanced so that this construction will preserve semantic equivalence.

6.2 Operational Semantics of SMIL making the Display Normal Form Semantically Equivalent

In this section, we briefly show how our operational semantics of SMIL can be enhanced so that any SMIL construction with a specified structure and its display normal form are semantically equivalent. We are deliberately brief due to space limitations and the fact that our operational semantics will need significant enhancements to incorporate other syntactic constructs of SMIL. First we close timed display sets under finite concatenations and re-interpret SMIL semantics with respect to them.

Definition 7 (Algebra of Timed Display Sets 1: Downward Closure and Concatenation)

1. Suppose tdi-1=(<type src="xx", ..dur=d1, attributeTestSecurity="y", T-begin1,T-end1>, {y}) and tdi-2=(<type src="xx", ..dur=d2, attributeTestSecurity="y", T-begin2, T-end2>, {y}) are two timed display units with the same source, attributeTestSecurity values, security components satisfying T-end1=T-begin2. Then we say that tdi-3=(<type src="xx", ..dur=d1, attributeTestSecurity ="y", T-begin1,T-end2>, {y}) is the concatenation of tdi-1 and tdi-2. We denote the concatenation of tdi-1 and tdi-2 by tdi-1;tdi-2.
2. We say that a timed display set TDS is concatenation closed if tdi-1,tdi-2εTDS => tdi-1;tdi-2εTDS.
3. We say that a timed display set TDS is downward closed if . =(<type src="xx", ..dur=d1, attributeTestSecurity="y", T-begin1,T-end1>, {y}) ε TDS, then =(<type src="xx", ..dur=d1, attributeTestSecurity="y",T-begin1',T-end1'>,{y}) εTDS for any T-begin'>T-begin and T-end' < t-end.

According to definition 7, downward closure allows any timed display set to include all segments of already included media streams. Concatenation closure allows piecing together successive segments of the same stream to obtain longer streams.

Lemma 2 (Minimal Concatenation Downward Closure of a Timed Display Set) CD Closure

1. Given a timed display set TDS, the concatenation closure of TDS, TDS* is defined as follows:
   TDS^0={(<type, src"x",attTestValue=Y>,t,t,{Y}) |(<type, src"x", attTestValue =Y>,t1,t2,{Y})
   e TDS and t1≤t≤t2}
   TDS^1=TDS
   TDS^{n+1}=TDS^n ;TDS
   TDS*= ∪{TDS^n | 0≤n}
2. TDS^=={(<type, src"x",attTestValue=Y>,t1,t2,{Y}) |(<type, src"x", attTestValue =Y>,t3,t4,{Y})
   e TDS and t1≥t3 and t4≤t2}

Then, (TDS*)^ is the minimal timed display set containing TDS that is both concatenation and downward closed.

Proof: Omitted

We now enhance the semantics of SMIL by using CD closure sets of base sets. Hence, we strengthen definition 5 as follows.

Definition 8 (Enhanced Semantics for SMIL)

Suppose S is a SMIL specification and [[ ]] is a basis mapping for the basic media elements B of S with the formal parameter t. Then we inductively extend [[ ]]+ to S as follows.

1. [[Null]]+ = Φ.
2. [[S']]+ = ([[S']])*^ for all basic media streams S' of S.
3. [[<seq> S1 S2</seq>]]+ = ([[S1]]+ U [[S2]]+(end([[S1]]+)/t))*^
5. [[<par> S1 S2</par>]]+ = [[S1]]+ U [[S2]]+.
6. [[<switch> S1 S2 </switch>]]+ = [[S1]]+ if S1 satisfies the attribute of the switch.
   = [[S2]]+ otherwise if S2 satisfies the attribute of the switch.
   = Φ otherwise.

We now say that the enhanced mapping [[ ]]+ is a semantic mapping parameterized by t.

Now we show how this semantics preserves the display normal form. Notice that the difficulty of the semantics given in definition 6 was with respect to piecing together successive segments of the same stream. By taking concatenations, this problem was solved in definition 6. Downward closures were taken to permit taking all subintervals of permitted streams.

Lemma 3 (Equivalence of Display Normal Form)

The two specifications shown in Figure 4 have the same semantics.

Proof: (Informal)

First observe that if S1 is the specification given on the left and S2 is the specification given on the right, then tdi ε [[S1]]+ iff tdi^n ε [[S2]]+. The reason being that S2 executes S1 arbitrarily many times. But, [[S2]]+ is concatenation and downward closed. Therefore, tdi^n ε [[S2]]+ iff tdi ε [[S2]]+. The reader will now see that downward closure was required in definition 8 in order to obtain tdi ε [[S2]]+ from tdi^n ε [[S2]]+.

6.3 Dynamic Runtime Activity.

As explained, any given SMIL specification S for surveillance is statically translated into its MLS normal form MLSNF(S). Then, when the runtime provides D, MLSNF(S) is translated into its display normal form, say DNF(MLSNF(S),D). Then the runtime takes each the set of streams within the switch that has duration of D, evaluates the switch, and depending upon the mode encrypts and transmits either the streams corresponding to normal operating mode or those that correspond to the emergency operating mode. The figure 5 shows the display normal form for the SECRET VIEW and briefly discusses mode evaluation procedures.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<customAttributesMODE>
       <customTestMode="Normal" title="Normal Mode"
         defaultState="true" override="hidden"
         uid="ControllerChoice" />
       <customTestMode id="Emergency" title="Emergency Mode"
         defaultState="false" override="hidden"
         uid="ControllerChoice" />
</customAttributesMODE>
<customAttributesSecurity>
       <customTestSecurity id="TS" title="Top-Secret"
         defaultState="true" override="hidden"/>
       <customTestSecurity id="S" title="Secret"
         defaultState="true" override="hidden"/>
       <customTestSecurity id="UC"  title="Unclassfied"
         defaultState="true" override="hidden"/>
</customAttributesSecurity>
  <body>
    <switch>
     <ref src="ModeNClassS.smil"    customTestMode ="Normal"  customTestSecurity ="S" />
    <ref src="ModeNClassS.smil"    customTestMode ="Emergency" customTestSecurity ="S" />
   </switch>
  </body>
</smil>
```

Figure 5: The Secret View of the SMIL Document of Figure3 in Display Normal Form.

The setting of the MODE value associated with a customTest MODE is as follows:

- The initial setting is taken from the value of the defaultState attribute, if present. If no default state is explicitly defined, a value of **false** is used.

- The URI (Controller Choice) defined by the uid attribute is checked to see if a persistent value has been defined for the custom test attribute with the associated id (Normal, Emergency). If such a value is present, it is used instead of the default state defined in the document (if any). Otherwise, the existing initial state is maintained.
- As with predefined system test attributes, this evaluation will occur in an implementation-defined manner. The value will be (re) evaluated dynamically, as described above

6.4. The Confidentiality Enforcing Encryption Model

Mobile handheld viewing devices [19] that have embedded SMIL players are the recipients. A smartcard, which enforces access control, is embedded into the display device [10]. Each display device has a unique smartcard depending on the classification of the guard that utilizes it and his classification and also any other rules set by the controller. A decryption key associated with the privileges of the guard is also embedded in the smartcard.

When a display device receives an encrypted SMIL document, the smartcard decrypts the appropriate segment depending on the available decryption key. We encrypt each view in the document as shown n Figure 6 with a unique Symmetric Key using the standard XML Encryption Specification [17]. An inbuilt Cryptix Parser that is programmed in firmware (or in software) to handle the decryption process would enable selective decryption of the appropriate view based on the access privileges as defined in the smartcard.

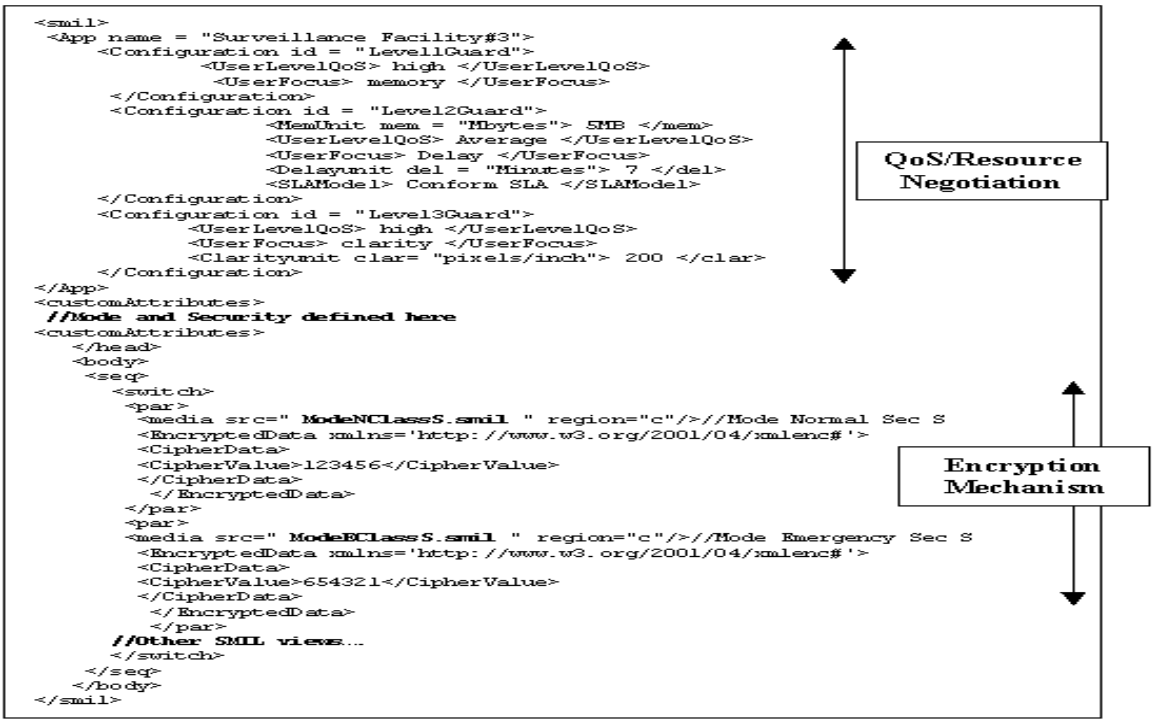

Figure 6: Display view with QoS and Encryption Parameters.

The Figure 6 depicted above shows the QoS Negotiation TAGS in accordance with HQML [25] and the Encryption tags applied to the display normal form of the secret view[Fig 5]to achieved fidelity and confidentiality. With encryption, we guarantee that nobody tampers the stream in transit even if there is mediate stream acquisition.

7. CONCLUSIONS

We have provided a model for audio-video surveillance of multi-level secured facilities during normal and pre-envisioned emergencies. We enhanced the SMIL specification with security decorations in order to achieve our goal of being able to satisfy MLS constraints during normal operations and provide controlled declassification during emergencies. Then we showed how to transform such a SMIL composition to its MLS normal form that preserve runtime semantics intended by SMIL constructs while creating views compliant with MLS requirements. Given the delay characteristics of a runtime, we show how to transform a SMIL document in MLS normal form so that the operating mode can be switched with the minimum delay while respecting runtime semantics of SMIL. Our ongoing work extends this basic framework to incorporate richer multimedia semantics as well as diverse security requirements such as no-reputable of media evidence, two way media channels and incorporate them in SMIL metamodels. Finally, this paper focuses on confidentiality issues. However, it is important to address data integrity and source authentication issues. These issues, along with the development of a prototype system are part of our future work.

REFERENCES

[1] B. K. Schmidt "An Architecture for Distributed, Interactive, Multi-Stream, Multi-Participant Audio and Video". Technical Report No CSL-TR-99-781, Stanford Computer Science Department.

[2] D. Wijesekera and J.Srivastava, "Quality of Service Metrics for Multimedia" in Multimedia Tools and Applications, Vol 2, No3 1996, pp. 127-166.

[3] J. Ayers et al. "Synchronized Multimedia Integration Language (SMIL 2.0)". World Wide Web Consortium (W3C). http://www.w3.org/TR/smil20/ (August 2001).

[4] A.G. Stoica and C. Farkas. "Secure XML Views" in *Proc. IFIP WG11.3 Working Conference on Database Security*, King's College, Cambridge, England.

[5] E.Bertino,M.A. Hammad ,W.G. Aref and A.K. Elmagarmid "An access control model for video database systems" in Conference on Information and Knowledge Management, 2000 .

[6] A. Gabillon, E. Bruno. Regulating Access to XML documents. in *Proc. IFIP WG11.3 Working Conference on Database Security*, Niagara on the Lake, Ontario, Canada, July 15-18, 2001

[7] E. Bertino, E. Ferrari S. Castano “Securing XML Documents with Author-X” in IEEE Internet Computing, vol 5,no3 May/June 2001

[8] E. Bertino, M. Braun, S. Castano, E. Ferrari, M. Mesiti. "AuthorX: A Java-Based System for XML Data Protection". In Proc. of the 14th Annual IFIP WG 11.3 Working Conference on Database Security, Schoorl, The Netherlands, August 2000

[9] E. Bertino, S. Castano, E. Ferrari and M. Mesiti. "Specifying and Enforcing Access Control Policies for XML Document Sources". World Wide Web Journal, vol. 3, n. 3, Baltzer Science Publishers.

[10] N.Kodali, D.Wijesekera“Regulating Access to SMIL formatted Pay-per-view Movies” in Workshop on XML Security, 2002.

[11] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Securing XML Documents," in Proc. of the 2000 International Conference on Extending Database Technology (EDBT2000), Konstanz, Germany, March 27-31, 2000..

[12] E. Damiani, S. De Capitani di Vimercati, E. Fernandez-Medina, P. Samarati “An Access Control System for SVG Documents” in Proc. IFIP WG11.3 Working Conference on Database Security.

[13] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati "XML Access Control Systems: A Component-Based Approach" in Proc. IFIP WG11.3 Working Conference on Database Security, Schoorl, The Netherlands, August 21-23, 2000.

[14] E.Damiani, S. De Capitani di Vimercati “Securing XML-Based Multimedia Content” to appear in SEC 2003, Atehns, Greece.

[15]The Triclops Camera at http://www.ptgrey.com/products/triclopsSDK/triclops.pdf

[16]Alpha works Suite :XML http://www.alphaworks.ibm.com/xml

[17] D.Eastlake et al “XML Encryption Syntax and Processing” at http://www.w3.org/TR/xmlenc-core/

[18] Spymake Integrated Surveillance Tools at http://www.spymakeronline.com/catalogue/surveillance.html

[19] Mobile VCMS™ - Field Data Collection System at http://www.acrcorp.com:8080/acr_vcms/mobile

[20] E. Ekudden, U.Horn, M.Melander and J.Olin“On-demand mobile media—A rich service experience for mobile users” Erricson.com White papers.

[21] Video Surveillance and Monitoring (VSAM) Homepage at http://www-2.cs.cmu.edu/~vsam/

[22] D. Elliot Bell and Leonard J.LaPadula Secure computer systems: Mathematical foundations. Technical Report 2547 (Volume I), MITRE, March 1973.

[23] Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E. “Role-based access control models” In IEEE Computer, 1998

[24] S. Jajodia, P. Samarati, V. S. Subrahmanian, ``A logical language for expressing authorizations," Proc. IEEE Symp. on Security and Privacy, Oakland, Calif., May 1997, pages 31-42

[25] X.Gu, K.Nahrstedt, W.Yuan, D.Wichadakul and D.Xu "An XML-based Quality of Service Enabling Language for the Web" in UIUCDCS-R-2001-2212 April 2001

[26] K. Mulmuley "Full Abstraction and Semantic Equivalence," ACM Doctoral Dissertation Award 1986, The MIT Press, Cambridge. MA, London, England, 1987

[27] G.D. Plotkin "A Powerdomain Construction" in SIAM Journal of Computing, Volume 5, pages 452-487.