ENFORCING INTEGRITY IN MULTIMEDIA SURVEILLANCE†

[2]Naren B. Kodali, [3]Csilla Farkas and **[1]Duminda Wijesekera**
[1]Center for Secure Information Systems, [2]George Mason University. [3]Department of Computer Science and Engineering, University of South Carolina..

Abstract: In this paper, we propose a secure distribution model for multimedia surveillance data, where different locations of the monitored facilities may have different security requirements. We propose a multilevel security framework, wherein the surveillance data streams are classified according to the sensitivity of the location of the surveillance devices, and users (guards) have corresponding security clearances. Guards monitor live multimedia feeds emanating from surveillance devices placed throughout the facility. Our position is that during normal mode of operation, guards are allowed to access only those multimedia streams for which they have the proper authorizations. However, in an emergency, guards may receive pre-computed emergency instructions and/or cover stories. We show how to compose multilevel secure SMIL documents, compute views for each security classification, enforce integrity, confidentiality and access control, and deliver the secure views to handheld devices while

Keywords: Physical surveillance, SMIL, Secure multimedia, Multi Level Security

1. INTRODUCTION

Modern physical surveillance and monitoring systems depend on electronic instruments [12,13,14] where monitored areas such as command and control centers and missile storage facilities of a military base or traffic controller units of airports

† This work was partially supported by the National Science Foundation under grants CCS-0113515 and 0112874.

are accessible only to predefined groups of people. It naturally follows that disclosure of live surveillance records of such facilities must follow the access restrictions of the physical facilities. For example, in an airport location, passengers and employees can enter common areas, like ticketing and waiting areas. However, secured areas, like luggage transport, are available for airport employees only. The highest security areas, such as the air traffic control room, are accessible to specialized personnel only. Our aim is to ensure that people who are not authorized to access a physical location should not be able to view the surveillance data of that location. We also ensure that the integrity of the data is preserved during transit. However, during emergency operations controlled dissemination of sensitive data may become necessary in order to obtain support services and/or to prevent panic. It has been shown, that during crisis people require clear instructions for cooperation. 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 dual-mode operation calls for issues of integrity to be consummately addressed. This paper proposes a framework to do so using appropriately secured *Synchronized Multimedia Integration Language (SMIL)* [3] formatted multimedia compositions.

Our proposal is to integrate monitoring and communication in a secure surveillance system that enforces access control restrictions on datwhile providing maximum availability. Our main contribution is the development of a framework to express multimedia compositions with their rich runtime semantics, techniques to enforce integrity and access control, and the use of cover stories to disseminate relevant material to users with lower clearance levels during emergencies. For simplicity, we assume a multilevel security classification of physical areas and their corresponding surveillance data. People accessing these facilities have varying levels of 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 appropriately cleared. We enforce that requirement during normal operations for guarding personnel. The main difference between a traditional Multilevel Secure (MLS) system and MLS for live surveillance feeds is that surveillance systems require the appropriate dissemination of classified information. 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 subject with lower levels of clearances.

Our model provides an integrated solution to manage surveillance devices and to collect audiovisual evidence for forensic analysis. We use an XML-based multimedia composition language 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 [12,14]. Second, mobile communication and usage of XML formatted data becomes faster and more prevalent [11,12] than in the past. Third, toolkit support is available to integrate XML compliant services [10,13,14], Therefore, with the right technological framework, our solution is portable to a wide range of general-purpose mobile multimedia devices such as those available in automobile navigation systems and hand-held PDA's.

The multimedia community has its own XML-like language known as SMIL [3] to compose presentations using stored and/or live sources. Unlike XML tags, SMIL associates explicit meaning to some of its tags, which is crucial to ensure that the semantics of the captured scene is appropriately presented at the display. Further, human perception associates a meaning to integrated audio-video sequences. Maintaining data integrity is crucial in conveying the intended meaning. We enrich existing proposals [4,8] for securing multimedia data by incorporating operational semantics of information. We show how non-emergency and emergency operations of a MLS facility can be composed as a SMIL document enriched with proposed extensions. We take such a composition and construct views authorized for different security classes. We refer to these constructs as MLS normal forms of a SMIL document with appropriate security decorations. Then, given the runtime characteristics of an operational platform, we show how to generate an view appropriate for that runtime, which we call a display normal form of a SMIL document. We then encrypt media streams and transmit them to intended recipients under normal and emergency operating conditions.

The rest of the paper is organized as follows. Section 2 provides our running example used to explain the application scenario and our solution. Section 3, describes related work. Section 4 describes SMIL. Section 5, describes the required static preprocessing, runtime activities including encryption and resource management are explained in Section 6. Section 7 concludes the paper.

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

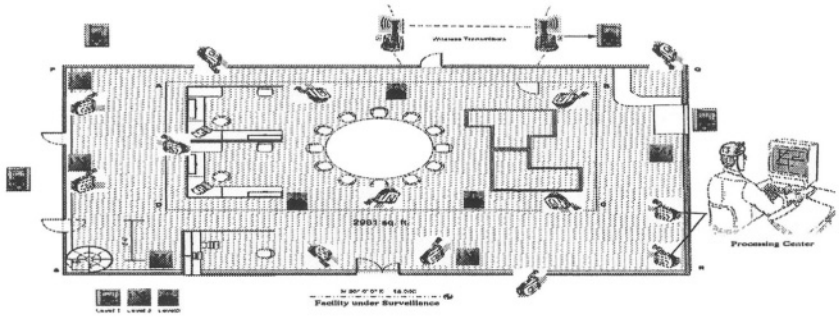

Figure 1: An Example MLS-Secure Facility

PQRS is UC. Employees, guards, support services personnel, and general public have $TS \bullet S \bullet UC$ clearances, where $\bullet$ 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 secret or unclassified area, but not vice versa. For electronic surveillance purposes, cameras and microphones are situated throughout the facility. Multimedia streams emanating from these devices are used to continuously 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.

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. MLS systems provide controlled information flow based on the security classification of objects (e.g., data items) and subjects (e.g., users) of the MLS system (e.g., applications running in behalf of a user) where data is allowed to flow only from low security levels to higher security levels [15]. 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 [15,16]. Regulating access to XML formatted text documents has been actively researched in the past few years offering many solutions. Bertino et al. [4,5], have

developed Author-X, a Java based system to secure XML documents that enforces access control policies at various granularities and corresponding user credentials.

Damiani et al. [6,7] developed an access control model where the tree structure of XML documents is exploited using XPATH expressions at different levels of granularity. They also propose an access control model [7] with complex information filters using stereo video graphics (SVG) to render maps of physical facilities for controlled dissemination of sensitive data within an image. Bertino at al. [4] 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. The proposed model provides a general framework of the problem domain, although no explanation is offered as to how access control objects to be released are formalized and enforced.

4. 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. SMIL has syntactic constructs for timing and synchronization. In this section, we explain those SMIL constructs, that are relevant and how they can be used to specify a multimedia document satisfying the application needs stated in Section 2.

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. The *<switch>* construct allows one to switch between 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 stereo.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 encodes the operating mode (normal or emergency) and the second encoding the security level of streams (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.

Figure 2 shows a simplified example of a SMIL specification for surveillance. The custom attribute *customTestMode* has values “Normal” and “Emergency”. Since the value of *customTestMode* is *hidden,* this attribute in each corresponding stream cannot be changed. The second part of the file consists of a switch statement where media streams connected by <par> constructs. Notice that the <switch> statement consists of two sections where first begins with the line <par customTestMODE= “Normal”> and the second begins with the line <par customTestMODE= “Emeregency”>. That specifies that the streams inside be shown under normal and emergency operating conditions. 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 TS 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. STATIC PREPROCESSING TO ENFORCE MLS

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.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<customAttributes>MODE>
        <customTestMode="Normal" title="Normal Mode"
          defaultState="true" override="hidden"
         <customTestMode id="Emergency" title="Emergency Mode"
          defaultState="true" override="hidden"
<customAttributes>MODE>
<customAttributes>Security>
        <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"/>
</customAttributes>Security>
<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(Unclassified)
<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>
```

Figure2: SMIL Specification of Figure 1

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 values 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. 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) or a 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 [17] designed to be used in XML formatted texts and databases. 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. Here, 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 SMIL document into its MLSNF.

Algorithm 1:toMLSNF (Conversion to a Normal Form)

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>
3. Similarly when, S is <par>S1 S2</par> , and S is <switch>S1 S2</switch>

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.

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

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 is standard, if we have a formal operational semantics for SMIL. While providing such semantics is not difficult, it does not exist yet. Therefore, while we are in the process of developing formal semantics for SMIL, we provide a preliminary operational semantics for the purposes of showing that our algorithms work as expected.

5.1 A Preliminary 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 [18]. Following customary practices in programming language semantics, our operational semantics and the proof of semantic equivalence is inductive in nature. 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 $\leqq$ T-end are arithmetic expressions of a single real variable $\mathbf{t}$ 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.

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) A set M of basic media streams with time durations 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. To the best of our knowledge, the informal semantics given the SMIL specification is abstractly captured by our operational semantics provided we can 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.

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 3 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.

6. RUNTIME BEHAVIOR OF THE MLS SYSTEM.

```
<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="videol" dur="45s"/>
<audio src="TSCameral.wav"  />
</par>
<par>
<video src="TSCamera2.rm" channel="videol" />
<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="CoverstoryTSl.rm" channel="videol" id="TSCoverstoryl"/>
<audio src="CoverstoryTSl.wav"  />
</par>
<par>
<video src="CoverstoryTSl.rm" channel="videol" id="TSCoverstoryl"/>
<audio src="CoverstoryTSl.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.

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 guard with top secret, secret, and unclassified clearances. In addition, in order to respond to emergencies, these specifications have a mode switch encoded using the custom attribute *attributeTestMode.* As seen in Figure 3, this attribute is to be evaluated at the beginning of a <switch> statement. That is unsatisfactory for intended purposes, because 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 the next section, we show how to rewrite a SMIL document with one <switch> statement for changing a mode to another SMIL document that evaluates the *attributeTestMode* at regular intervals. Although in theory any system could switch its operating mode in an arbitrarily small time intervals, practical considerations limits this interval. This minimum switching granularity may depend upon many parameters such as hardware, software. 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 Informal Display Normal Form

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 Figure 4).

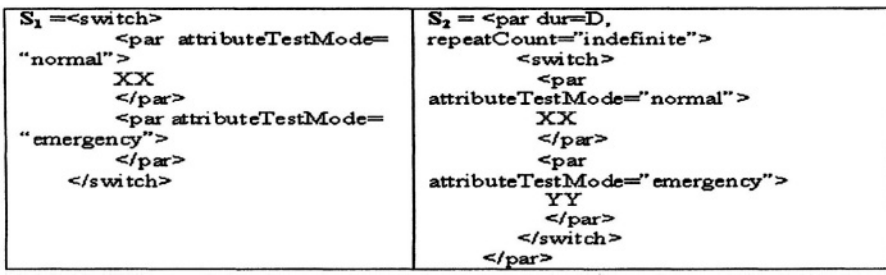

Figure 4: Mode Evaluation Semantics.

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. Our translation into display normal form is not semantically equivalent under semantics provide in definition 6. However, we intend to enhance the semantics so that this construction will preserve semantic equivalence in our future work.

6.2 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 initial setting of the mode is taken from the value of the defaultState attribute. If no default state is explicitly defined, a value of **false** is used. The URI (Controller Choice) is checked to see if a persistent (Normal, Emergency) is defined instead of the default. 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 and Integrity Enforcing Encryption Model

Mobile handheld viewing devices [14] that have embedded SMIL players are the recipients. A smartcard, which enforces access control, is embedded into the display device [9]. Each display device has a unique smartcard depending on the classification of the guard that utilizes it and his classification and 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 in Figure 6 with a unique Symmetric Key using the standard XML Encryption Specification. 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.

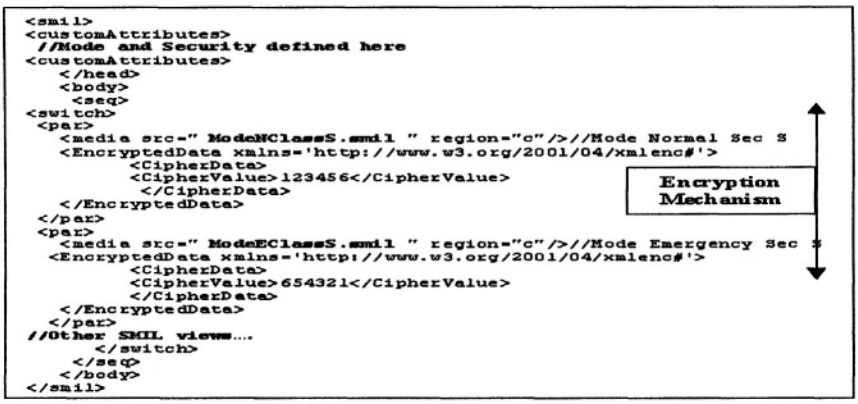

Figure 6: Display view with QoS and Encryption Parameters.

The Figure 6 depicted above shows the encryption tags applied to the display normal form of the secret view [Figure5] to achieve 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 to satisfy MLS constraints during normal operations and provide controlled declassification during emergencies while maintaining the integrity and confidentiality of the data in transit. Then we showed how to transform such a SMIL composition to its MLS normal form that preserves 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 non-repudiation of media evidence, and incorporate them in SMIL metamodels. Finally, this paper focuses on confidentiality issues. However, it is also important to address source authentication issues, which 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] E.Bertino, M.A. Hammad ,W.G. Aref and A.K. Elmagarmid "An access control model for video database systems" in Conference on Info and Knowledge Management, 2000

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

[6] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Securing XML Documents," in Proc. of EDBT2000), Konstanz, Germany, March 27-31, 2000..

[7] E. Damiani, S. De Capitani di Vimercati, E. Fernandez-Medina, P. Samarati "An Access Control System for SVG Documents" in Proc. IFIP WG11.3 , DBSEC '02.

[8] E.Damiani, S. De Capitani di Vimercati "Securing XML-Based Multimedia Content" Security and privacy in the Age of Uncertainty, Pages 61-72.

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

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

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

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

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

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

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

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

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

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