SECRETS: A Secure Real-Time Multimedia Surveillance System *

Naren Kodali[1], Duminda Wijesekera[1,2] and Csilla Farkas[3,4]

[2]Center for Secure Information Systems, [1]Dept of Info. and Software Eng., George Mason University, Fairfax, VA 22030–4444, [3]Information Security Laboratory, [4]Dept of Computer Science and Eng., University of South Carolina,Columbia, SC-29208
email:{nkodali|dwijesek}@gmu.edu,farkas@cse.sc.edu

Abstract. We propose a surveillance framework (*SECRETS*: SECure Real-time ElecTronic Surveillance) that is a practical solution to safeguarding sensitive physical facilities like command and control centers, missile storage facilities of a military base, traffic controller units of airports and other high-volume public areas by providing controlled secure distribution of live multimedia data recorded on-site onto display devices with different access permissions in a multi-level secure environment. Our methodology 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 recipients with corresponding security classifications receive live media streams, but during emergencies, they can receive pre-computed emergency instructions and/or cover stories. This facilitates avoiding disorder and obtaining appropriate support from individuals with lower levels of clearances. We use SMIL [Aya01]formatted, multimedia feeds including cover stories, and explain how *SECRETS* can compose necessary multimedia documents, compute views for each security classification, enforce access control and deliver media to the handheld devices while respecting both wealthy run time semantics [KFW03b]of multimedia as well as MLS security [KFW03a].

1 Introduction

The concept of safeguarding facilities by monitoring has been widely implemented with the use of electronic surveillance instruments [Spy,VCM]. In general monitored areas accessible only to predefined groups of people and therefore the disclosure of live surveillance records to the same set of people, even when they are remotely located is a logical continuation.

Consider an airport in which passengers and employees can enter common areas, like transportation facilities, and waiting areas. However, secured areas,

* This work was partially supported by the National Science Foundation under grants CCS-0113515 and IIS-0237782.

like luggage transport and service stations, are available for authorized employees only. The highest security areas, such as the air traffic control room, are accessible to specialized personnel who are appropriately authorized. The keyword here is "authorization", meaning that people who are not authorized to access a physical location should not be allowed physical or electronic access to that location. In the surveillance world, the exact same rules apply and the potential recipient of the surveillance data must have the same authorization that an ordinary person of any trade would have to be physically or electronically present at 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 system, that we call *SECRETS* that can be practically deployed in the aforementioned situation using appropriately secured SMIL (Synchronized Multimedia Integration Language) formatted multimedia compositions. A detailed deployment infrastructure along with its compliance with present day standards, COTS products and commercially available authoring and display devices is provided.

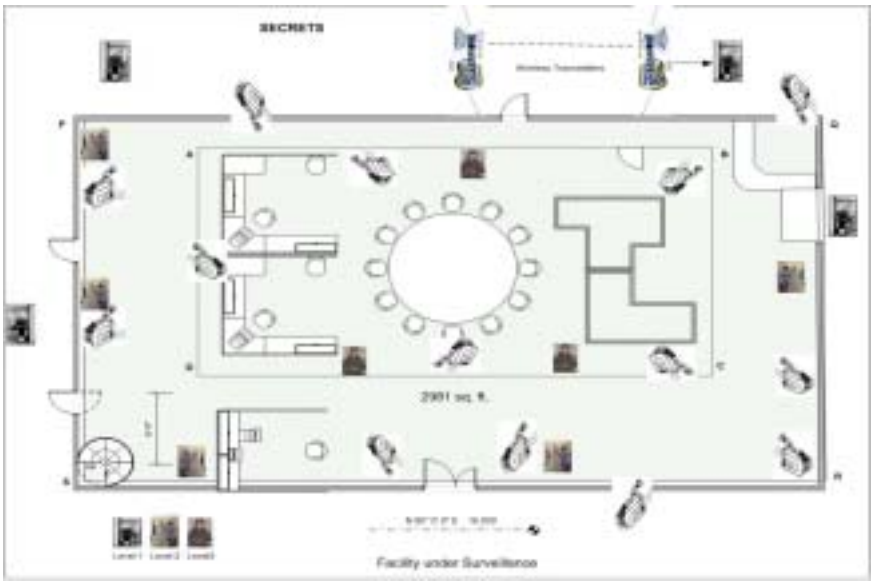


Fig. 1. A hypothetical facility under Surveillance with *SECRETS*

The primary motive of *SECRETS* is to integrate monitoring and communication units in a secure surveillance system that enforces access control restrictions on data while providing maximum availability. Our main contribution is the development of a system 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 on guarding personnel during normal operations. 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.

SECRETS provides an integrated solution to manage surveillance devices, and to assert and collect audiovisual evidence for forensic analysis as well as to improve the quality of cover stories. We use SMIL-based multimedia composition framework adapted for multi-level physical surveillance. The reason for selecting SMIL is guided by recent industrial trends. First, many browsers and off-the-shelf display and communication devices are becoming SMIL compliant [VSA,Nok]. Secondly, as mobile communication, using SMIL, becomes faster and more integrated, mobile devices are becoming available [Spy,VCM]. Thirdly, toolkit support is becoming available to integrate XML compliant services [PCV02,Bul98]. Therefore, with the right technological framework, our solution becomes portable to a wide range on-site surveillance activities.

We extend upon existing proposals for securing textual documents [DdVPS00] and [DdVPS02] 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 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 presents some related work, 3describes SMIL, 4discusses the architecture and general requirements of a general problem domain. In 5, we present preprocessing fundamentals in the MLS domain and the associated normal form.Section 6 and Section 7 deal with the runtime operations and real-time deployment issues respectively. Section 8 concludes the paper.

2 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 [Sch99]. In this architecture, multimedia streams originating from multiple sources can be combined to provide media clips that accommodate look-around capabilities.

SMIL has been the focus of active research [RvOHB99,RHO99,SSC00], and many models for adaption to real world scenarios have been provided. A release control for SMIL formatted multimedia objects for pay-per-view movies on the Internet that enforces DAC is described in [KW02]. The cinematic structure consisting of acts, scenes, frames of an actual movies are written as a SMIL document without losing the sense of a story. Here access is restricted to the granularity of an *act* in a movie. A secure and progressively updatable SMIL document [KWJ03] is used to enforce RBAC and respond to traffic emergencies. In an emergency response situation, different recipients of the live feeds have to be discriminated to people playing different roles.

Multilevel security (MLS) has been widely studied to ensure data confidentiality, integrity, and availability . MLS systems provide controlled information flow(from higher level to the lower level) 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). Damiani et al [DdV03] also discuss feature protection of XML format images. Its primary focus is controlled dissemination of sensitive data within an image. 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 [BHAE02] provide 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.

While most models addresses the need of multimedia, their approach does not incorporate semantics of multimedia. None of the 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. 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.

Independent of security, Quality of Service (QoS) is an integral part of multimedia. Wijesekera et al. [WS96] proposed properties of quality metrics associated with continuous media and Gu et al. [GNY+01] propose *HQML*, a language to negotiate some QoS parameters between clients and server.

3 SMIL: Synchronized Multimedia Integration Language

SMIL [Aya01] is an extension to XML developed by W3C to author multimedia presentations with audio, video, text and images to be integrated and synchronized. The distinguishing features of SMIL over XML are the syntactic constructs for timing and synchronizing live and stored media streams with qualitative requirements. In addition, SMIL provides a syntax for spatial layout including non-textual and non-image media and hyperlinks. We do not address the later aspects of SMIL in this paper. Consequently we explain those SMIL constructs that are relevant for our application.

SMIL constructs for synchronizing media are ⟨ seq⟩, ⟨ excl ⟩ and ⟨ par⟩. They are used to hierarchically specify synchronized multimedia compositions. The ⟨ seq⟩ element plays its children one after another in sequence. ⟨ excl⟩ specifies that its children are played one child at a time, but does not impose any order. The ⟨ par⟩ 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.

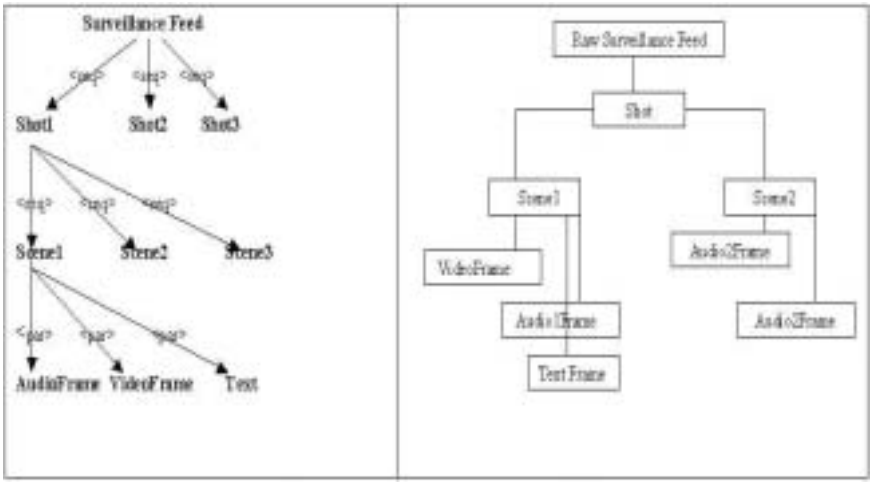


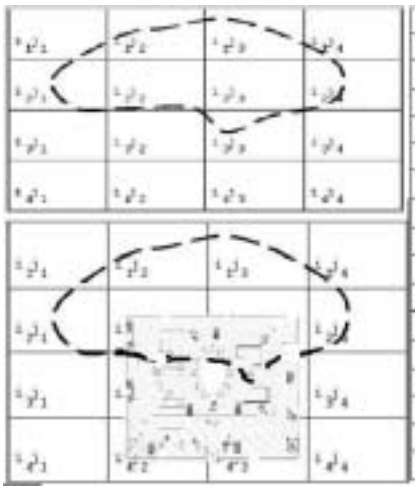
Fig. 2. Representation of SMIL Constructs

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 equal active durations, SMIL provides many constructs to equate them. 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 and the latter specifies a master-slave relationship between synchronized streams. 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.

4 *SECRETS* Architecture

Figure 1 shows a hypothetical military facility with varying levels of sensitivity based on geographic location. 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. 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 > S > UC clearances, where " > " 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 and are regulated based on geographical co-ordinates as shown in 3.



	Co-ordinate	Classification	Camera
1	i_1j_1	Unclassified	UC Camera 1
2	i_1j_2	Unclassified	UC Camera 2
3	i_1j_3	Unclassified	UC Camera 3
4	i_1j_4	Unclassified	UC Camera 4
5	i_2j_1	Secret	S Camera 1
6	i_2j_2	Secret	S Camera 2
7	i_2j_3	Secret	S Camera 3
8	i_2j_4	Secret	S Camera 4
9	i_3j_1	Top-Secret	TS Camera 1
10	i_3j_2	Top-Secret	TS Camera 2
11	i_3j_3	Top-Secret	TS Camera 3
12	i_3j_4	Secret	S Camera 5
13	i_4j_1	Unclassified	UC Camera 5
14	i_4j_2	Unclassified	UC Camera 6
15	i_4j_3	Unclassified	UC Camera 7
16	i_4j_4	Unclassified	UC Camera 8

Fig. 3. Geographic grid representing capture locations

4.1 Information flow in *SECRETS*

The capture and the ensuing transfer of sensitive information within *SECRETS* is regulated via an Client-Server architecture as shown in Figure 4. Multimedia streams emanating from strategically located devices are continuously used are continuously captured and transmitted to a centralized control facility for dissemination and then directed to handheld devices of appropriate security personnel.

As we observe in Figure 4 The server consists of a Camera Manager and an Audio Manager which are integrated based on a system clock with the help of a synchronization manager. These empirical needs of synchronization and integration are also the foundations of the SMIL language. The repositories of captured information and cover stories is resident in a database that communicates with the synchronization manager via the buffer manager during the preprocessing and formatting of real time data in to SMIL documents. The transfer is handled by Network Manager and could be wired(if within the facility) or through wireless radio links. The Network manger as an dual interface at both the client and the server locations to facilitate secure and effective communication.

In *SECRETS* the clients could be stationary on-site personnel or guards equipped with display devices and performing duties at remote locations. At the client location there are audio and video managers to enable the stream integration based on the system clock. The access/release control of the documents is controlled

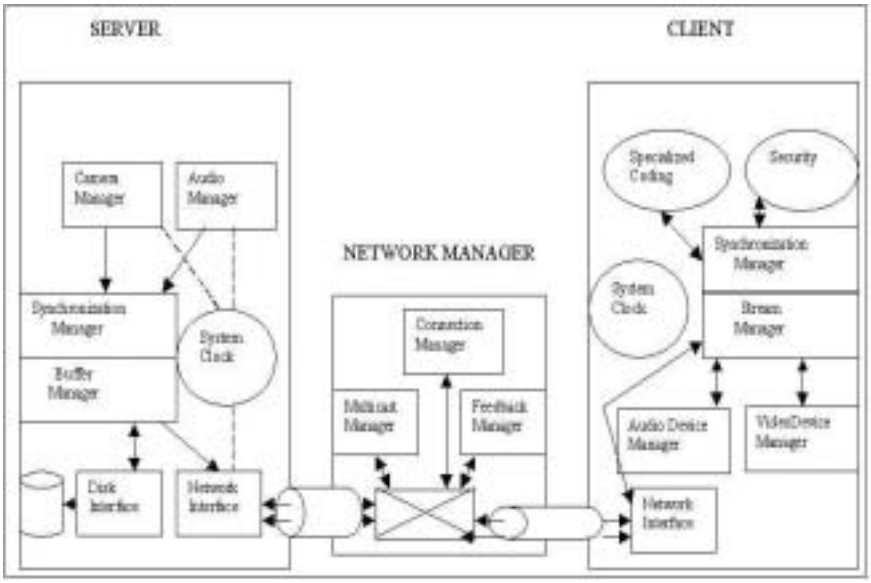


Fig. 4. Black-box Architecture of Secrets

via specialized coding or hardware devices, collectively called *smartcards* within the display device that can interpret the encryption and release only the allowed views corresponding to the privileges of the client.
An inbuilt Cryptix Parser that is programmed in firmware (or in software) exists within the smartcard to handle the decryption process and enable selective decryption of the appropriate view based on the access privileges as defined in the smartcard. All transmissions of media files to various people in the hierarchy use standard hypertext transfer protocol (HTTP) web protocol or the Wi-Fi(802.11) wireless transfer protocol. The encryption mechanism to enforce integrity is discussed in detail in Section 7

5 Multi-Level Secure Operations

Any security architecture needs the proper definition of the *subject* and the *protection object*. In Multi Level Security each access permission is determined by the security clearance of the subject and the security classification of the accessed object. Security labels form a lattice structure with the dominance relation among the labels. Information flow between the security labels is controlled based on the security objectives. In *SECRETS* the information flow is from low security objects to high security objects, that is, from a dominated object to a dominating object. Assuming that our access permissions are "read" permissions, it means that a subject is allowed to access an object only if the subject's security clearance dominates the security classification of the object.

We now formally define MLS and how constraints could be used to construct the access control lists.
Dominance relation is a partial order, that is: Given security labels l_1, l_2, l_3 : Reflexive: $\forall l_1, l_1$ dominates l_1, Transitive: $\forall l_1, l_2, l_3$, if l_1 dominates l_2 and l_2 dominates l_3 then l_1 dominates l_3 and Antisymmetric: $\forall l_1, l_2$ if l_1 dominates l_2 and l_2 dominates l_1 then $l_1 = l_2$.

To model the dominance relation, first we construct the transitive closure of dominance relations. Then, we use this closure to identify the security objects in the *normal form* of a specification S that are dominated by the security clearance of the subject. Let $Class(s)$ denote the classification of subject s. L denotes the lattice structure and binary relation $dominates(l_1, l_2)$, $l_1, l_2 \in L$ denotes that label l_1 dominates label l_2. To generate all labels dominated by the security classification a subject ($Class(s)$), we

1. Generate transitive closure of dominance relation
2. Let$Dominated(s) = \emptyset$
3. For all pairs $dominates(l_i, l_j)$,
   where $l_i = Class(s)$, $Dominated(s) = Dominated(s) \cup l_j$

To permit accesses for a subject to objects in mlsNF, we use the set *Dominated* to determine the appropriate data items. That is, a subject is granted the access a to an object o if the security clearance of the subject dominates the security classification of the object.Therefore MLS could be stated as an (s,o,a) triple. $\forall s$, if $Class(s)$ and $\{l_{i_1}, \ldots, l_{i_n}\} \in Dominated(s)$ and $o \in Cl_{i_k}$ $k = 1, \ldots, n$ then (s, o, a).

5.1 MLS SMIL fragment with Security attributes

The sample SMIL fragment shown below is a simplified example derived from a complex SMIL specification using the defined attributes. 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 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= "Emergency"⟩. 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 . 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 main composition is encoded using a ⟨switch⟩ statement that is to be switched based on the operating mode (normal or emergency).

```
<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="Unclassified"
        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(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)
    </body>
</smil>
```

5.2 Normalform and Operational Semantics

The MLS Normal Form of a SMIL specification S as the one in Section 5.1 caaled *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. We now give a formal definition of mlsNF.

Definition 1 (MLS Normal Form) *We say that a SMIL specification (s) is in the mlsNF(MLS Normal Form) if it is of the form* $\langle$ *seq* $\rangle$ $\langle$ *par* $\rangle$ $C_ts(s)\langle$ */par*$\rangle$ $\langle par\rangle$ $C_s(s)\langle$ */par*$\rangle$ $\langle$ *par* $\rangle$ $C_u(s)$ $\langle$ */par* $\rangle$ $\langle$ */seq* $\rangle$ *where all Security classifications in* C_ts *(s),* $C_s(s)$*,* C_u *(s)are respectively Top-Secret, Secret and Unclassified.*

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. A detailed discussion of the *normal form*,the algorithm for conversion of an arbitrary SMIL fragment to its mlsNF, the operational semantics based on the normal form and its proof of correctness can be obtained from our previous papers [KFW03a,KFW03b].

6 Runtime Operations

In order to respond to emergencies, the SMIL specifications have a mode switch encoded using a custom attribute attributeTestMode. As observed in the SMIL fragment, this attribute is to be evaluated at the beginning of a ⟨switch⟩ statement. This is unsatisfactory for the intended surveillance purposes, because the operating mode could vary many times after the switch is initially evaluated. If 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 fire-fighting 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 as discussed in the next section.

6.1 Informal Display Normal Form

The following SMIL specification given below, has the same structure as the fragment considered in Section 5.1. 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 fragment S1 can be translated as shown in S2.

```
S1 =<switch>
<par  attributeTestMode= "normal"> XX </par>
<par attributeTestMode= "emergency"></par>
</switch>

S2 = <par dur=D, repeatCount="indefinite"><switch>
<par attributeTestMode="normal"> XX</par>
<par attributeTestMode="emergency">YY </par>
</switch>
</par>
```

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 attributeTest-Mode specified in the form where the switch is reevaluated every D time units is said to be in display normal for the attribute attributeTestMode and time duration D. We can now informally say that every SMIL document where the attributeTestMode 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 limited operational semantics. However these semantics can be enhanced so that this construction will preserve semantic equivalence.

6.2 Secure View Generation

To construct a view corresponding to a security classification, for any given SMIL specification S for we need to statically preprocess and translate S 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). Given below is a the secure view constructed for the *secret* security classification for both the normal and emergency modes using the procedure described above.

```
<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"
       -------
</customAttributesMODE>
<customAttributesSecurity>
  ------
</customAttributesSecurity>
  <body>
    <switch>
<ref src="ModeNClassS.smil"
customTestMode ="Normal" customTestSecurity="S"/>
<ref src="ModeNClassS.smil"
customTestMode ="Emergency" customTestSecurity="S"/>
   </switch>
  </body>
</smil>
```

The *SECRETS* runtime takes each the set of streams within the switch that has duration of D, evaluates the switch, and depending upon the mode encrypts

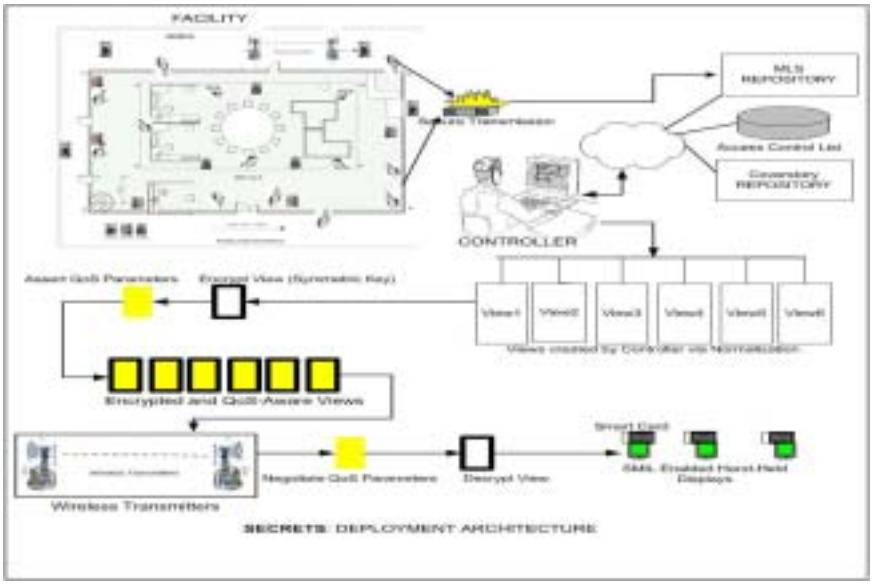


Fig. 5. View Generation and Deployment Architecture

and transmits either the streams corresponding to normal operating mode or those that correspond to the emergency operating mode. The mode evaluation procedures for setting of the MODE value associated with a customTest MODE is as follows:

1. 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.
2. 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.
3. As with predefined system test attributes, this evaluation will occur in an implementation-defined manner. The value will be (re) evaluated dynamically.

6.3 Quality of Service Issues

The SLA (Service Level Agreement determines the specifications and restrictions that have to be communicated between the client and the server in order to maintain good quality [WS96]. The requirements of the processors and memory (primary and secondary), and other technicalities such as tolerable delay, loss, pixels have to be negotiated prior or sometimes during the transfer process. HQML [GNY+01] proposes an XML based language for the exchange of processor characteristics. The most important characteristic is the amount of buffer, in

terms of memory that the recipient device should have in order to maintain continuity. These specifications would be represented within the SMIL document, so that the recipient device will first prepare or disqualify itself for a reception. In *SECRETS*, the QoS parameters are generally negotiated prior to the display. They could be set as custom defined attributes that have to resolve to true for the display to happen. We could use some of the standard attributes of the switch statement systemRequired, systemScreenDepth, and systemScreenSize to enforce regulation.

```
<App name = "Surveillance Facility#3">
     <Configuration id = "Level1Guard">
             <UserLevelQoS> high </UserLevelQoS>
              <UserFocus> memory </UserFocus>
      </Configuration>
      <Configuration id = "Level2Guard">
                  <MemUnit mem = "Mbytes"> 5MB </mem>
                  <UserLevelQoS> Average </UserLevelQoS>
                  <UserFocus> Delay </UserFocus>
                  <Delayunit del = "Minutes"> 7 </del>
                  <SLAModel> Conform SLA </SLAModel>
     </Configuration>
     <Configuration id = "Level3Guard">
            <UserLevelQoS> high </UserLevelQoS>
            <UserFocus> clarity </UserFocus>
            <Clarityunit clar= "pixels/inch"> 200 </clar>
     </Configuration>
</App>
```

The HQML fragment above describes QoS declaration and negotiation. The mobile or stationary device on receiving this file should be able to make decisions based on the user/server defined thresholds.

7 Client-Server Deployment

Mobile handheld viewing devices that have embedded SMIL players are the recipients in *SECRETS*. A smartcard, which enforces access control, is embedded into the display device [KW02,KWJ03]. 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 thereby effectively transferring the load from the server onto the recipient device. When a display device receives an encrypted SMIL document, the smartcard decrypts the appropriate segment depending on the available decryption key. We use XML Encryption for encrypting the views as well as transferring the keys, embedded in the SMIL document. as represented below . An inbuilt Cryptix Parser handles the decryption process would enable selective decryption of the appropriate

view based on the access privileges as defined in the smartcard. All transmissions of media files to various people in the hierarchy use standard hypertext transfer protocol (HTTP) web protocol or the Wi-Fi(802.11) wireless transfer protocol. In the SMIL document, each view is encrypted with an unique SymmetricKey depending on the security classification and the process is repeated for all the views within the document. All the encrypted views have a corresponding symmetric decryption key (which is the same as the encryption key) and the recipient smartcard. Figure 6 summarizes the details of the process and provides a run-time algorithm for *SECRETS*.

```
<smil>
 -----
<switch>
<par>
 <media src= "ModeNClassS.smil" customTest3 = "Emergency"/>
      <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'>
      <CipherData>
      <CipherValue>654321</CipherValue>
      </CipherData>
      </EncryptedData>
 </par>
 <par> <media src=" ModeNClassTS.smil" customTest3="Normal"/>
       ------
 </par>
//Other SMIL views.
       ------
 <\smil>
```

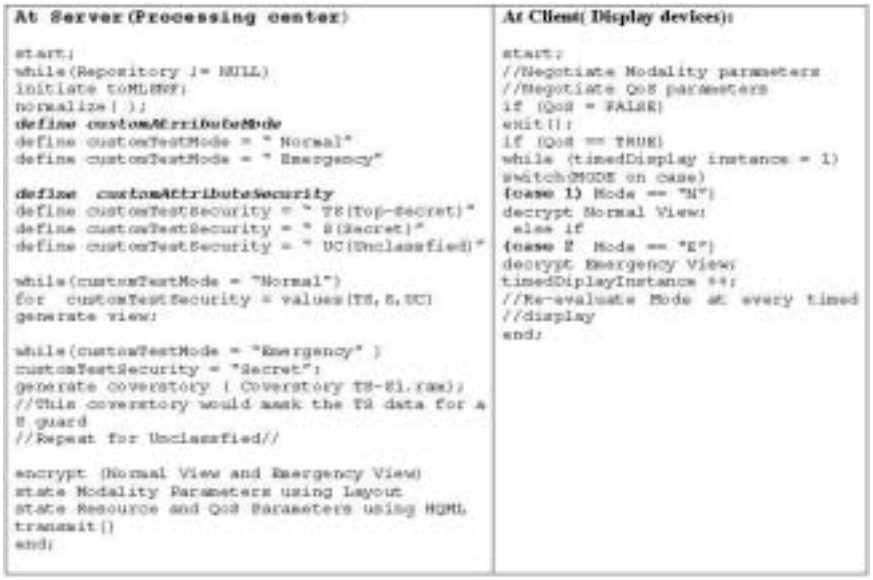
The fragment above represents the encryption procedure embedded in SMIL and is explained as below

1. The granularity of encryption in SCERETS is a view
2. The Symmetric key cipher is (3DES CBC)
3. The Symmetric key has an associated Class based on the intended recipient " SymmetricKey Class [TS,S,UC] ".
4. CipherData contains a CipherReference, a reference which helps in the transformations necessary to obtain the encrypted data

The Figure 5 pictorially represents the process of generating and encrypting the views and the black-box architecture for deployment in *SECRETS* that has been discussed in the Sections 6 and 7.

8 Conclusions

We have presented *SECRETS*, a surveillance framework 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



```
At Server(Processing center)

start;
while(Repository != NULL)
initiate toMLSNF;
normalize( );
define customAttributeMode
define customTestMode = " Normal"
define customTestMode = " Emergency"

define  customAttributeSecurity
define customTestSecurity = " TS(Top-Secret)"
define customTestSecurity = " S(Secret)"
define customTestSecurity = " UC(Unclassfied)"

while(customTestMode = "Normal")
for  customTestSecurity = values(TS, S, UC)
generate view;

while(customTestMode = "Emergency" )
customTestSecurity = "Secret";
generate coverstory ( Coverstory TS-S1.raw);
//This coverstory would mask the TS data for a
S guard
//Repeat for Unclassfied//

encrypt (Normal View and Emergency View)
state Modality Parameters using Layout
state Resource and QoS Parameters using HQML
transmit()
end;
```

```
At Client( Display devices):

start;
//Negotiate Modality parameters
//Negotiate QoS parameters
if (QoS = FALSE)
exit();
if (QoS == TRUE)
while (timedDisplay instance = 1)
switch(MODE on case)
{case 1) Mode == "N"}
decrypt Normal View;
 else if
{case 2  Mode == "E"}
decrypt Emergency View;
timedDiplayInstance ++;
//Re-evaluate Mode at every timed
//display
end;
```

Fig. 6. *SECRETS* Run-time Algorithm

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 non-repudiation of media evidence, two-way media channels and incorporate them in SMIL metamodels. Finally, this paper focuses on confidentiality issues. However, it is also important to address data integrity and source authentication issues. These issues, along with the development of a complete and comprehensive prototype system are part of our future work.

References

[Aya01] Jeff Ayars. *Synchronized Multimedia Integration Language*. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-smil20-20010807.

[BHAE02] Elisa Bertino, Moustafa Hammad, Walid Aref, and Ahmed Elmagarmid. An access control model for video database systems. In *Conferece on Information and Knowledge Management*, 2002.

[Bul98] David Bulterman. Grins: A graphical interface for creating and playing smil documents. In *Proc. of Seventh Int'l World Wide Web Conf. (WWW7)*. Elsevier Science, New York, April 1998.

[DdV03] Ernesto Damiani and Sabrina De Capitani di Vimercati. Securing xml based multimedia content. In *18th IFIP International Information Security Conference*, 2003.

[DdVPS00] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Securing XML documents. *Lecture Notes in Computer Science*, 1777:121–122, 2000.

[DdVPS02] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. A fine grained access control system for xml documents. *ACM Transactions on Information and System Security*, 5, 2002.

[GNY+01] Xiaohui Gu, Klara Nahrstedt, Wanghong Yuan, Duangdao Wichadakul, and Dongyan Xu. An xml-based quality of service enabling language for the web, 2001.

[KFW03a] Naren Kodali, Csilla Farkas, and Duminda Wijesekera. Enforcing integrity in multimedia surveillance. In *IFIP 11.5 Working Conference on Integrity and Internal Control in Information Systems*, 2003.

[KFW03b] Naren Kodali, Csilla Farkas, and Duminda Wijesekera. Multimedia access contol using rdf metadata. In *Workshop on Metadata for Security, WMS 03*, 2003.

[KW02] Naren Kodali and Duminda Wijesekera. Regulating access to smil formatted pay-per-view movies. In *2002 ACM Workshop on XML Security*, 2002.

[KWJ03] Naren Kodali, Duminda Wijesekera, and J.B.Michael. Sputers: A secure traffic surveillance and emergency response architecture. In *submission to the Journal of Intelligent Transportaion Systems*, 2003.

[Nok] Mobile Internet Toolkit: Nokia. www.nokia.com.

[PCV02] Kari Pihkala, Pablo Cesar, and Petri Vuorimaa. Cross platform smil player. In *International Conference on Communications, Internet and Information Technology*, 2002.

[RHO99] L. Rutledge, L. Hardman, and J. Ossenbruggen. The use of smil: Multimedia research currently applied on a global scale, 1999.

[RvOHB99] Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, and Dick C. A. Bulterman. Anticipating SMIL 2.0: the developing cooperative infrastructure for multimedia on the Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1421–1430, 1999.

[Sch99] B. K. Schmidt. An architecture for distributed, interactive, multi-stream, multi-participant audio and video. In *Technical Report No CSL-TR-99-781, Stanford Computer Science Department*, 1999.

[Spy] Spymake. Integrated surveillance tools http://www.spymakeronline.com/.

[SSC00] Paulo Nazareno Maia Sampaio, C. A. S. Santos, and Jean-Pierre Courtiat. About the semantic verification of SMIL documents. In *IEEE International Conference on Multimedia and Expo (III)*, pages 1675–1678, 2000.

[VCM] Mobile VCMS. Field data collection system http://www.acrcorp.com.

[VSA] VSAM. Video surveilance and monitoring webpage at http://www-2.cs.cmu.edu/ vsam/.

[WS96] Duminda Wijesekera and Jaideep Srivastava. Quality of service (qos) metrics for continuous media. *Multimedia Tools and Applications*, 3(2):127–166, 1996.