Chapter 9

Secure Audio Databases

This chapter presents three models that can be used for the development of a secure audio database and its surrounding components. These three models should allow an entity to limit what is to be received and heard by a requesting entity with a given security classification. Mixing or adding an audio clip to a requested digital audio file will then be briefly discussed. Various ways of reducing suspicion for an assembled audio file will be presented, after which some security issues concerning the presented models will be highlighted.

9.1 Introduction

As already mentioned in chapter 8, the models presented in this chapter follow a similar format to the secure image/video database models. The basic aim of the database interface component remains the same as before, namely to collect all the authorisation information regarding (in this case) a digital audio file. Similarly, the SMI component's main responsibilities remain the same, namely to ensure that only the authorised parts of an audio file is transmitted to a requesting entity. The main aims for the three phases are also the same as the previous models even though the actual details change to cater for audio files.

The three models that will be presented in this chapter are: the basic audio editing model, the frequency/amplitude editing model, and the audio-effect editing model. These will now be further discussed.

9.2 The secure audio database models

The same model structure will be used for all three of the following secure audio database models; what differentiates these models is the actual information that is gathered and recorded within the security objects, and how this information is used to assemble an audio file according to a given security classification. The model structure illustrated in figure 9.1 below therefore applies to all three of the models (the differences will be explained in detail under each section).

In this model, an *audio object* is created instead of an image or video object. The audio object includes the security information associated with a time range within the original audio file. It is therefore the type of data that is embedded within these audio objects and how this data is used within the three phases (together with audio editing algorithms) that forms the difference between the three models. In the below model illustration, the variable n is equal to the number of audio objects that are defined for that specific audio file. As with the previous models, the solid

and dashed arrows represent the flow of information between the components of the model, and dashed lines are used for the special cases where a request is being made between two components.

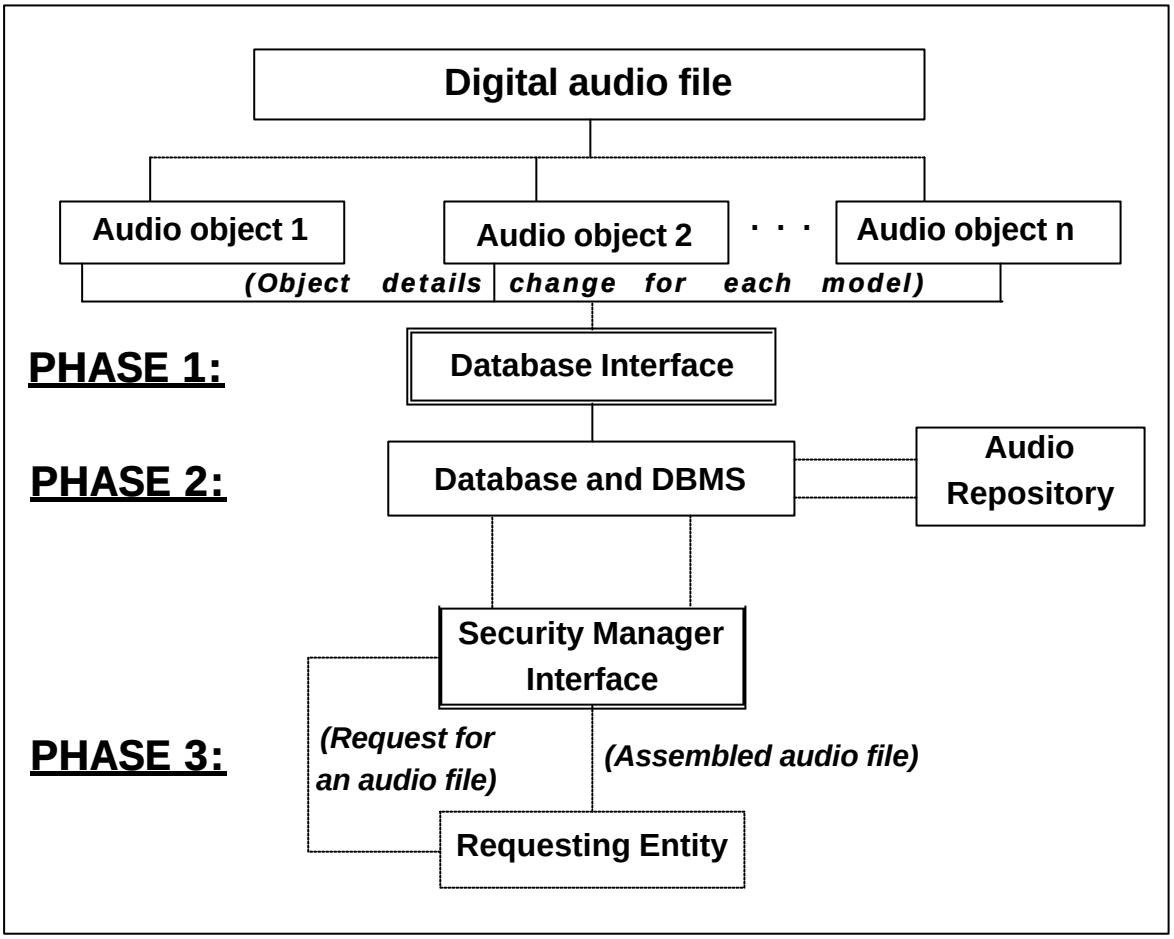

Figure 9.1: The model structure for all three secure audio database models

9.2.1 The basic audio editing model

This model is very similar in theory to the basic frame editing model discussed in chapter 8. The basic audio editing model allows an entity to select a time range within a digital audio file, preferably measured in milliseconds, and assign a classification to this selection (in contrast to selecting a frame range using the basic frame editing model). A request for an audio file using this model prevents certain time ranges to be included in the assembled audio file for entities with a security classification below the associated audio object's classification.

An authorised entity must first create all the audio objects for a particular digital audio file. The database interface or a client program that securely communicates with the database interface will be used to create the audio objects (as with the

previous models, only the database interface is allowed to write and modify data on the audio database). The entity performs this task by scanning through the entire audio stream and by selecting the start and end times of the ranges that are to be associated with an audio object. Each audio object must then be assigned a classification. The time ranges within the audio file that have not been classified can be accessed by any authenticated requesting entity.

The first phase is complete after the entity has finished creating all of the audio objects. The database interface must then write all the security objects to the audio database, and the original audio file to the audio repository. The audio file will be assigned a unique identifier, which will be stored in the audio database together with the path of the corresponding audio file within the repository. The time ranges for each audio object together with the associated classification will be written to the database. Every audio object must contain the unique identifier of its associated audio file as a reference to it.

Once all this data has been recorded, the RDBMS is prepared to receive requests from the SMI for that audio file. As with previous models, all audio file requests must be performed using the SMI or a client program that securely communicates with the SMI. When requested, the RDBMS will transfer the original audio file together with the associated audio objects to the SMI. On receipt, the SMI will scan through the list of audio objects and remove all parts of the original audio file associated with an audio object whose classification is above the requesting entity's security classification. After these parts have been removed, the rest of the original file is assembled and transferred to the requesting entity.

The audio files can also be pre-calculated in this model. After all the audio objects for a particular audio file have been created, the database interface will assemble an audio file for each possible security classification in that computer system. The database interface will, for each classification, scan through the created audio objects to see which objects are associated with a higher classification. The associated time ranges for these unauthorised audio objects will then be removed from the original file. This process is complete when an audio file has been pre-calculated for each security classification.

The pre-calculated audio files are then stored in the audio repository and an entry containing the path name and associated classification of each of these files will be recorded in the audio database. This variation saves time by allowing the SMI to respond to a request for an audio file without assembling it each time; a request for the correct pre-calculated audio file is all that is needed (assuming that it has

been correctly assembled). Pre-calculating audio files for the audio editing model is not essential since it is relatively quick to simply remove parts of an audio file as required by this model. Avoiding this allows an entity to change audio object properties directly on the database (using the database interface), instead of reassembling one or more pre-calculated audio files each time a modification is required (a combination of the two approaches is also possible).

An entity can now be sure that only the authorised time ranges of an audio file is transferred (as one file) to a requesting entity. An entity may often want an audio file to be made available with only certain parts being dubbed out depending on the classification of a requesting entity e.g. bad language may need to be removed from the original file for users that are below a certain age (assuming that a user's age is related to his/her assigned security classification).

In similarity to the basic frame editing model, this model forces an entity to classify an entire time range to be removed if unauthorised; it does not allow associations to be made at a finer (signal) level. It may be the case that only certain sounds within a time range need to be removed from the audio file, or that a man's voice must be altered to sound like a woman's voice for entity's with a particular security classification. These issues will be covered in the following two models.

9.2.2 The frequency/amplitude editing model

As mentioned in chapter 4, an analogue signal's frequency determines the pitch of a sound, and the amplitude determines the volume of a sound. These two sound properties allow us to eliminate high pitched or low pitched sounds by analysing an audio file's frequencies, or to remove sounds above or below a certain volume by analysing the amplitude. An entity may for example want to remove all frequencies within an audio file that are out of the range of a telephone, which can transmit signals in the region of 300 Hz to 3300 Hz (Shay, 1995), in order to make it believable that the audio file originated from a telephone conversation (i.e. when entities of a particular security classification make a request for that audio file, the assembled file must be changed in this manner).

This model allows an entity to look for certain frequencies or amplitudes within an audio file and associate this information with an audio object (together with the time range for which signals of that frequency/amplitude must be removed if unauthorised). This approach therefore allows an entity to specify logical access control at a deeper (signal) level, as opposed to the basic audio editing model

which only allows entire time ranges to be removed. This audio object acts as a filter, ensuring that only certain ranges of frequencies/amplitudes are allowed to 'pass through' to the assembled audio file (just like an equalizer attached to a sound system filters out treble or bass sounds).

The audio objects are created as before using the database interface. This time, an authorised entity must find which frequencies or amplitudes would be out of the desired range for a requesting entity with a particular security classification. A low pitched sound or a sound with a very high volume for example may need to be eliminated in these cases. Certain sounds may also have a constant frequency or amplitude throughout a time range; these sounds could therefore be removed by simply specifying that frequency/amplitude value. The database interface should include a search feature that automatically finds and selects a frequency or amplitude value or range within specified time range parameters. The entity must then be able to select which of these are to be associated with an audio object, and what classification must be assigned to that object.

Phase 2 of this model is similar to the basic audio editing model's storage phase. The main difference in phase 2 is that the frequency, amplitude, and associated time range information is written to the audio database, instead of just the time range information (together with the audio object's classification and linked audio file). Phase 3 requires more complex algorithms to be able to select and remove any range of frequencies and amplitudes associated with the unauthorised audio objects. This process will alternatively be performed in phase 1 if the audio files are pre-calculated for each possible security classification.

The frequency/amplitude editing model allows an entity more control in what is to be removed from an audio file if necessary. An entity can now remove certain types of sounds from an audio file without having to completely remove the entire time range within which that particular sound resides. This approach does however require more advanced audio editing algorithms to be used, which increases the computer system's processing requirements. Pre-calculating the audio files is therefore recommended in most cases (especially if frequency or amplitude editing is often required).

9.2.3 The audio-effect editing model

An unauthorised part of an audio file may not always need to be removed in order for the audio file to be considered authorised. It may be the case that a minor

modification to that part of the audio file is all that is needed for it to be considered authorised, regardless of whether that specific modification is actually required or whether it is simply an optional alternative to removing that part of the audio file. An example of this may be to change a man's voice to sound like a woman's voice for a certain part of the audio file (which can be done by increasing the pitch of the signal). This model allows an entity to specify which audio effects are to be applied on certain parts of an audio file if requested by an entity with a specific security classification.

In this model, the database interface must include a number of audio effects such as reversing audio, volume amplification, frequency editing, adding echo's, etc., which can be used to modify the original audio file when needed. An authorised entity must therefore know where in the audio file these changes must be made, select this time range, select an audio effect to be applied, and associate this information with an audio object. It may also be possible to automate the selection of some of these time ranges in certain cases e.g. if all sounds below a certain volume need to be increased by a certain amount, a search feature can be programmed to find and select these parts, allowing the entity to decide which of these must have an audio effect applied to it. Finally, the entity must assign a classification to each defined audio object.

Each available audio effect must have a unique identifier which will be used to associate the audio object's classification and time range with the audio effect to be applied if unauthorised. Each audio object is written to the database, while the original audio file is stored in the repository. At the time of a request, the SMI receives this information and applies the necessary audio effects to the time ranges associated with an unauthorised audio object. The number of available audio effects and their complexity depends on what potential modifications may need to be performed on an audio file. As with the region selection methods in the secure image database model, any number of audio effect algorithms may be added, as long as these changes are duplicated on both the database interface and the SMI. Naturally, this will not apply in cases where the audio files are pre-calculated in the first phase (in this case, the algorithms used during selection and assembly within this phase must remain the same).

The audio-effect editing model can be used to change parts of an audio file in order to make only certain requesting entities believe that these modifications existed in the original audio file. This model may require advanced audio editing techniques to be used, and so a powerful server may be needed to accommodate this (although this depends on the type of audio effects required).

9.3 Adding/mixing audio clips with original files

It may be the case that another audio file (or part of one) must be inserted within the original audio file when requested by an entity with a particular security classification. As with the secure video database models, this can be useful in cases where the requesting entity must believe that a fictional event (or sound) formed part of the original audio file. Alternatively, another audio file (or part of one) can be *mixed* together with specified parts of the original audio file, as opposed to inserting it within the original file.

To add an audio file, an authorised entity must select an insertion point within the original audio file and associate this with an *insertion audio object*. A unique identifier for the audio file to be inserted must also be associated with this audio object together with an assigned classification. A time range within the audio file to be inserted needs to be specified if only a part of this file must be added to the original file. The insertion audio object is written to the audio database together with the other audio objects, and the audio file to be inserted is stored in the audio repository together with the original audio file (naturally if the audio files are pre-calculated, this information does not need to be recorded).

Mixing an audio file with a part of the original file can be especially useful in adding background sounds e.g. making it seem as though a crowd is surrounding a person speaking at an event by mixing the sound of a crowd together with that part of the original audio file. To mix two files together, an authorised entity must choose a time range within the original audio file and associate this with a *merging audio object*. This audio object must be linked with the audio file that is to be mixed together with the original file, and the start position within this audio file must also be associated. Finally, the merging audio object is assigned a classification. This information is written to the audio database and repository unless the audio files are pre-calculated for each possible security classification.

9.4 Suspicion

Note that a different approach to the one used in the previous models is needed in order to reduce suspicion after unauthorised parts of the original audio file have been removed. This is because a background image obviously cannot be used to replace these unauthorised parts in an attempt to ‘fill in the missing gaps’. An audio file will most likely only be heard and not graphically viewed (in waveform), especially for applications that request the audio file on behalf of a user and

automatically play the audio file when received (only after it has been assembled). A different approach that is more suitable to audio files therefore needs be used in order to try and reduce (and if possible eliminate) suspicion.

One of the best ways to reduce suspicion is to ensure that the unauthorised parts of the original audio file are perfectly removed during the assembly process. This is obviously related to the selection process in the first phase, since this process will be repeated at the time of assembly. The time ranges that are associated with an audio object should therefore be selected very accurately. This may be tricky (depending much on the case at hand); the audio editing software used in the selection (and reselection) process should therefore allow editing at a very fine (millisecond) level.

The entity creating the audio objects should also preview the assembled audio file for each security classification in order to ensure that a smooth transition takes place before and after a part of the original file has been removed, or a part of another file added. A smoothing operation such as fading in/out or frequency smoothing should be used if necessary to try and make a persuasive transition. This should also help to lessen suspicion in cases where the waveform is visibly scanned for any possible modifications.

Adding another piece of an audio file in place an unauthorised time range that must be removed could also help reduce the level of suspicion, provided that the addition fits within the context of that range (i.e. it must be appropriate). The reason for adding a part of an audio file to an original file could therefore simply be to help ‘cover up’ the deletion, thereby reducing suspicion. Alternatively, a part of the original audio file may be modified using available sound effects. For example, instead of directly trying to cover up the alteration, an authorised entity may try to make it seem as though something happened to that section of the audio file that was beyond the control of the audio file’s source, such as cellular phone distortions or the signal that was breaking up, forces of nature, or lagging in the network’s transmission (especially for real-time audio streams from a remote server). The entity must not however exaggerate any effects, since this may end up making the situation worse.

9.5 Security

As with the secure video database approaches, all the models presented in this chapter have similar security requirements to the secure image database model’s

security requirements. This is because the aims for the input, storage and output phases that make up the secure audio database models are the same as the aims for the previously presented model's three phases. All the security requirements regarding each of these 3 phases (as specified earlier) must therefore be ensured in order for a system based on any of these models to achieve its objective.

The general security requirements that must be ensured for these models depend on the actual computer system's environment and the value of the audio data, although certain requirements such as accurate identification and authentication must be ensured for all model implementations. The speed at which an audio file must be assembled and retrieved after its request must be considered when determining other less significant security requirements. The basic audio editing model should be relatively efficient when assembling the audio file, while the frequency/amplitude editing model and the audio-effect editing model can be quite processor intensive, depending on the algorithms used to assemble the audio file.

9.6 Conclusion

The models discussed in this chapter should allow an entity to limit what parts of an audio file will be received by a requesting entity. The audio editing model, although more efficient than the other models, can only be used to remove entire time ranges within the original audio file. The frequency/amplitude editing model allows an entity to restrict access at the signal level, allowing certain frequencies or amplitudes to be filtered out. The audio-effect editing model allows an entity to define which parts of an audio file should be modified using available sound effects, depending on a requesting entity's security classification.

Combining these three models should not be a problem, especially since the main differences lie in the actual content of the audio objects created for each model. The combined model's components must therefore be able to distinguish between the three different types of audio objects and perform the actions required by the applicable model. As with the secure video database model, requesting only a segment of the original audio file should be allowed by assembling only the time range as specified by the requesting entity.