*Journal of Information Security*, 2010, 1, 29-40
doi:10.4236/jis.2010.11004 Published Online July 2010 (http://www.SciRP.org/journal/jis)

# Design and Implementation of Multilevel Access Control in Synchronized Audio to Audio Steganography Using Symmetric Polynomial Scheme

**Jeddy Nafeesa Begum[1], Krishnan Kumar[1], Vembu Sumathy[2]**
[1]*Department of Computer Science and Engineering, Government College of Engineering, Bargur, India*
[2]*Department of Electronics and Communications Engineering, Government College of Technology, Coimbatore, India*
*E-mail*: {*nafeesa_jeddy*, *pkk_kumar*}*@yahoo.com*, *sumi_gct*2001*@yahoo.co.in*
*Received March* 5, 2010; *revised July* 15, 2010; *accepted July* 19, 2010

## Abstract

Steganography techniques are used in Multimedia data transfer to prevent adversaries from eaves dropping. Synchronized audio to audio steganography deals with recording the secret audio, hiding it in another audio file and subsequently sending to multiple receivers. This paper proposes a Multilevel Access control in Synchronized audio steganography, so that Audio files which are meant for the users of low level class can be listened by higher level users, whereas the vice-versa is not allowed. To provide multilevel access control, symmetric polynomial based scheme is used. The steganography scheme makes it possible to hide the audio in different bit locations of host media without inviting suspicion. The Secret file is embedded in a cover media with a key. At the receiving end the key can be derived by all the classes which are higher in the hierarchy using symmetric polynomial and the audio file is played. The system is implemented and found to be secure, fast and scalable. Simulation results show that the system is dynamic in nature and allows any type of hierarchy. The proposed approach is better even during frequent member joins and leaves. The computation cost is reduced as the same algorithm is used for key computation and descendant key derivation. Steganography technique used in this paper does not use the conventional LSB's and uses two bit positions and the hidden data occurs only from a frame which is dictated by the key that is used. Hence the quality of stego data is improved.

**Keywords:** Steganography, Multilevel Access Control, Synchronized Audio, Symmetric Polynomial, Dynamic, Scalable

## 1. Introduction

Transmission of audio files is very important for many applications and it is found that this transmission takes place through an insecure medium. For a live session where on the go audio transmission takes place, efficient techniques should be used. One way of preventing the dissemination of secret audio is through digital audio stenography. This protects valuable information from unauthorized persons. For real time audio transmissions the secret data is recorded and send subsequently to the receivers. For hiding the message there is a need for a secret key that is available with all receivers. This key has to be changed to preserve forward and backward secrecy. There is an additional very important requirement called the multilevel access control. There are many scenarios in which situation arises, that only some users should be able to hear the data or all higher level users should also be able to hear the message that are relayed to the descendant users. To implement such a multilevel access control in steganography symmetric polynomial approach is used. In most existing schemes, key derivation is different from key computation. Key derivation needs iterative computation of keys for nodes along the path from a node to its descendant, which is inefficient if the path is long. In this scheme, both operations are same by substituting (different) parameters in the same polynomial function assigned to node *v*. Thus, the key derivation efficiency can be improved. Our scheme also supports full dynamics at both node and user levels and permits any random access hierarchies. More importantly, removing nodes and/or users is an operation

 

as simple as adding nodes and/or users in the hierarchy. A trusted Central Authority (CA) can assign secrets (*i.e.*, polynomials) to corresponding nodes so that nodes can compute their keys. Also, nodes can derive their descendants' keys without involvement of the CA once polynomial functions were distributed to them. In addition, the storage requirement and computation complexity at the CA are almost same as that at individual nodes, thus, the CA would not be a performance bottleneck and can deal with dynamic operations efficiently.

The rest of the paper is as follows, Section 2 deals with related work Section 3 gives an insight into multilevel access control problem. Section 4 gives the system Overview, Section 5 describes the audio steganography method Section 6 deals about the symmetric polynomial approach Section 7 shows the simulation results and Section 8 gives the performance analysis and Section 8 concludes the paper.

## 2. Related Work

### 2.1. Related Work in Steganography

Information hiding using steganography [1] relates to protection of text, image, audio and digital content on a cover medium [2-5]. The cover media in many cases has been an image [1]. Aoki presented a method in which information that is useful for widening the base band is hidden into the speech data [6]. Sub band Phase shifting was also proposed for acoustic data hiding [3]. All these schemes focus on data that is stored in a hard disk or any other hardware whereas there are many applications like military warfare where the audio data is to be given in real time as in live broadcast system. Techniques for hiding the audio in real time came into existence [7] and systems for synchronized audio steganography has been developed and evaluated [8]. In our scheme secret speech data is recorded and at the same time it is sent to the receiver and a trusted receiver extracts the speech from the stego data using the key which is shared between the server and the receiver.

### 2.2. Related Work in Multilevel Access Control

The first multi level access solution was proposed by Akl *et al.* [9,10] in 1983 and followed by many others [11-21]. These schemes basically rely on a one-way function so that a node $v$ can easily compute $v$'s descendants' keys whereas $v$'s key is computationally difficult to compute by $v$'s descendant nodes. In this paper, we propose a new scheme based on symmetric polynomials for synchronized audio data. Unlike many existing schemes based on one-way functions, our scheme is based on a secret sharing method which makes the scheme unconditionally secure [21,22]. Also, this multilevel access control requires two types of key operations: (1) key computation. A node $v$ computes its own key and (2) key derivation. A node $v$ computes its descendants' keys.

## 3. Multi Level Access Control Problem

In practice, many group applications contain multiple related data streams and have the members with various access privileges. These applications prevail in various scenarios.

1) Multimedia applications distributing data in multilayer coding format. For example, in a video broadcast, users with a normal TV receiver can receive the normal format, while others with HDTV receivers can receive both the normal format and the extra information needed to achieve HDTV resolution.

2) Communications in hierarchically managed organizations, such as military group communications where participants have different access authorizations.

3) Multilevel access control can be effectively used in Audio library and patient monitoring system.

4) E-newspaper subscription service may have multiple data streams. The service provider classifies the users into membership groups and provides data streams according to the subscription.

5) Video multicasting service in which users can subscribe to services with different video quality.

Defense messaging systems where the server sends messages and one or more can see the message according to the access rights.

In these applications, group members subscribe to different data streams, or possibly multiple of them. Thus, it is necessary to develop group access control mechanism that supports the multi-level access privilege, which is referred to as the Multilevel Access Control.

## 4. System Overview

Multilevel Access Control applied to real time audio to audio steganography is useful for organizations which have a hierarchical structure. e.g., in the Indian Military system the following hierarchy exists in "**Figure 1**".

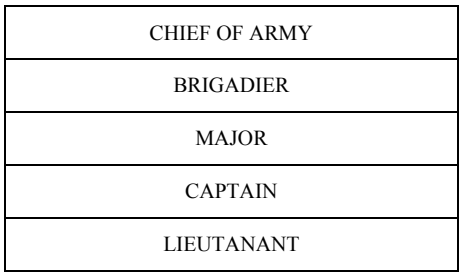


**Figure 1. Military hierarchy.**

 

In such a type of system, audio messages sent to a lower class should be heard by the active members of lower class and also by all active members of the higher class. It is not only essential to maintain the access control but the data should be hidden as well. The sequence of events is as follows.

*At the server*:

1) Generate a general polynomial.

2) Give a symmetric polynomial to each of the classes.

3) Record the real time audio on a microphone.

4) Use Steganography technique to hide the audio into another audio.

5) A text can also be hidden in an audio file.

6) The file is encrypted by the class key for whom the Message is to be relayed.

7) The symmetric polynomial generates a key in this case. The server takes care to include class dynamics so the hierarchy can be changed at any time.

8) Users can join or leave a class at all instances. Keys are recalculated so that Forward and Backward secrecy is maintained.

9) If the users within the group need to transfer message among themselves. The private key of the users is used.

The above steps are given pictorially in "**Figure 2**".

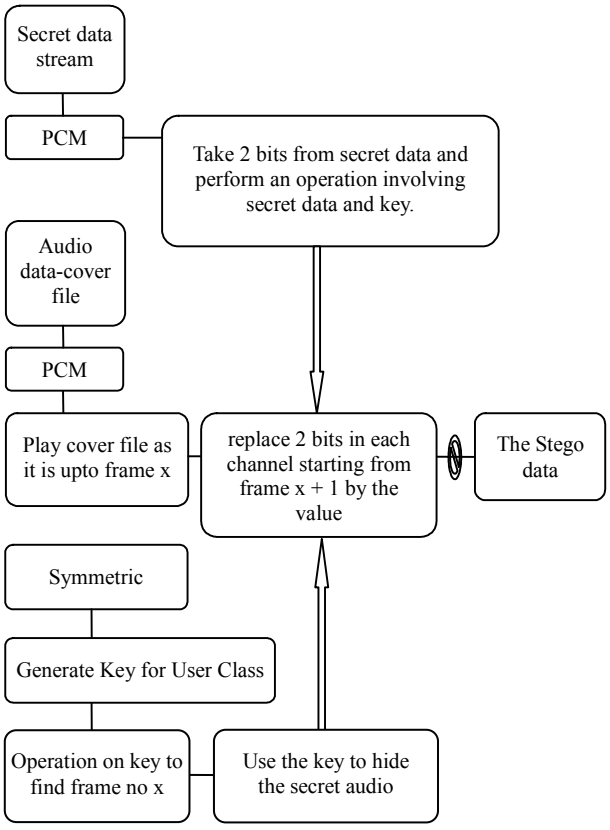


**Figure 2. At the server.**

*At the receiver*:

1) All the active receivers will receive the audio file.

2) If the recipient belongs to the actual intended class he can use the polynomial to get the hidden audio file instantaneously.

3) If the recipient belongs to a class lower than the Actual intended class in the hierarchy, he will not be able to derive the key .The polynomial derivation method will give a null value.

4) If the recipient belongs to a higher class he can derive the key and hear to the audio file and in case a text message was sent it can be seen.

5) The users at the same class can transfer messages among them.

6) When a user leaves or joins. The new polynomials are given by the server and the private keys also get updated according to the new polynomial. Other classes are not affected by this.

7) Service messages can be sent from higher class users to lower class users.

The above steps are explained pictorially in "**Figure 3**".

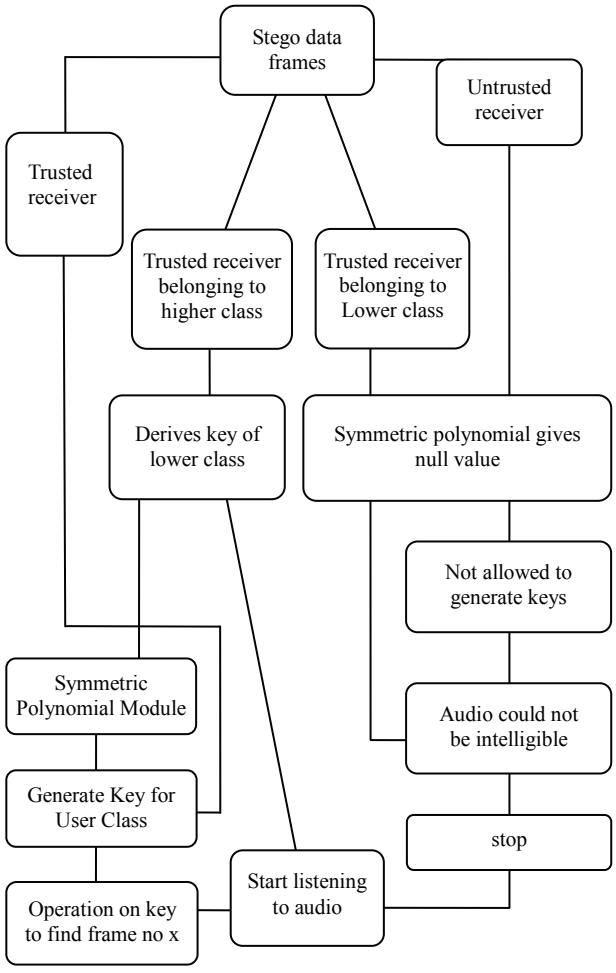


**Figure 3. At the receiver.**

 

### 4.1. Features of our Model

Solutions of a synchronized steganography have been given in past [8]. In this once the stego data reaches the destination the audio can be listened by the trusted receiver. Our contributions are

1) The key is used during the embedding process also.

2) The key is not a simple key it identifies a class of users.

3) If the key used belongs to a low level group in the hierarchy, the higher level class of user can derive the key using the symmetric polynomial approach and listen to it.

4) There can be normal message transfer among the Group elements and also service messages from higher classes.

5) Forward and backward secrecy is maintained.

6) It is a dynamic one where new hierarchies can be introduced, User level and class level dynamics are taken care.

## 5. Synchronized Audio to Audio Steganography

The data to be sent is not available. It is recorded in real time before starting the steganography scheme. When the covering media is being played at the same time the audio file is recorded and put into the cover file. The stego bit stream is then transmitted to the receivers. Multilevel Access control using symmetric polynomial is used at this stage to generate the key to make secure transmission of the audio file. According to the hierarchy the trusted users are able to retrieve the hidden audio file.In this system, both of cover data and secret data are divided into fix-sized frames according to pulse code modulation setting. To cover low size and high phonetic quality the sampling rate of the hidden audio is set to 8 kHz. Three main processes are involved in the synchronized audio to audio steganography.

1) Using data sampling, acoustic signals are embedded into another audio.

2) Bit Embedding: The key used helps in hiding the audio file in bit positions and once the bit positions are found data is hidden after performing an operation on secret data and the key.

3) Synchronized Process: Malicious and intentional attacks can be avoided as the secret data is real time.

### 5.1. Algorithm

Step 1: Record the Secret Speech Data: The audio files are divided into fix-sized frames and set to be specific PCM format. PCM quantification is decided by sampling rate, sampling size, and sampling channel. The PCM property of cover audio wav is set to be 32kHz-16bit-2ch, while the secret wav data is 16kHz-8bit-1ch.

Formula used for steganography:

Steganography process:

cover_medium + hidden_data + stego_key
= stego_medium

Part of The Wave File Format opened using Notepad:
52 49 46 46 24 40 01 00 57 41 56 45 66 6D 74 RIFF $ @.
Wave Fmt10 00 00 00 01 00 02 00 11 2B 00 00 44 AC 00 00 ……. + …D…04 00 10 00 64 61 74 61 00 40 01 00 00 00 00 00 …. data. @.....
Wave_Format_PCM: 01 11 channel count: 02 00
Samples: 11 2B 00 00 bytes: 44 AC 00 00
Block align: 04 00 bits per sample: 10 00

Step 2: Use Symmetric Polynomial to calculate key of the class

Step 3: Perform calculation and decide the frame from which the data is to be embedded.

Step 4: Decide two bit locations in each frame and clear the bit in the locations.

$$cmask1 = (2^{loc1} - 1) \text{ xor (keybit)}$$

$$\text{cmask2} = (2^{loc2} - 1) \text{ xor (Keybit)}$$

$$\text{cmask} = \text{cmask1} \wedge \text{cmask2}$$

hide the secret data bits into these bit locations by again performing an operation on the secret data along with the key. The cover media has two channels so data is written on both the channels. Other bits are not changed.

Step 5: The next set of data will go to the next frame.

Step 6: Do the repetitive process till the recoding is over.

Step 8: Transmit using sockets.

Step 9: At the receiving end, use the key and play the audio.

Step 10: If the receiving user belongs to higher class, he can derive the key and listen to the audio.

## 6. Symmetric Polynomial Approach

### 6.1. Symmetric Polynomial

A CA selects a large positive integer $P$ as the system modulus, $p$ need not be a prime and a threshold number $t$ so that less than $t + 1$ users cannot collaborate together to disclose their ancestors' keys. Then, the CA can randomly generate a symmetric polynomial in $m$ variables with co-efficient from $Z_p$ in which the degree of any variables is at most $t$ as:

$$P(x_1, x_2, \ldots, x_m) = \sum_{i_1=0}^{t} \sum_{i_2=0}^{t} \cdots \sum_{i_m=0}^{t} a_{i_1, i_2, \ldots, i_m} x_1^{i_1} x_2^{i_2} \cdots x_m^{i_m} \pmod{P}$$

where $a_{i_1, i_2, \ldots, i_m}$ are randomly generated coefficients by the CA. The polynomial function $P(x_1, x_2, \ldots, x_m)$ is kept as a secret to the CA. Every class in the hierarchy

 

has a polynomial function which is derived from $P(x_1, x_2, \ldots, x_m)$ and the polynomial function is transmitted to each class securely by the CA.

*Example for Symmetric Polynomial*

The following polynomial function is a suitable example for symmetric polynomials.

$$f(x_1, \ldots, x_n) = \sum_{i_1=0}^{w} \cdots \sum_{i_n=0}^{w} a_{i_1, i_2, \ldots, i_n} x_1^{i_1} \ldots x_n^{i_n}$$

where $a_{i_1, i_2, \ldots, i_n} = a_{\pi(i_1, i_2, \ldots, i_n)}$

For all permutations π of $\{1, \ldots n\}$

For example,

suppose $n = 3$ and $w = 2$,

Let $i_1$, $i_2$, $i_3$ be as follows

$a_{0,0,0} = 13$

$a_{0,0,1} = a_{0,1,0} = a_{1,0,0} = 3$

$a_{0,0,2} = a_{0,2,0} = a_{2,0,0} = 7$

$a_{0,1,1} = a_{1,0,1} = a_{1,1,0} = 4$

$a_{0,1,2} = a_{0,2,1} = a_{1,0,2} = a_{2,0,1} = a_{2,1,0} = 8$

$a_{0,2,2} = a_{2,0,2} = a_{2,2,0} = 9$

$a_{1,2,2} = a_{2,1,2} = a_{2,2,1} = 11$

$a_{2,2,2} = 5$

## 6.2. Polynomial Function

To derive proper keys in the hierarchy, the CA generates some publicly known numbers

1) $n$ random numbers $s_i$ associated with $C_i$ for $i = 1, 2, \ldots n$ and 2) and $(m - 1)$ additional random numbers $r_j$ for $j = 1, 2, \ldots, m - 1$

(Note: $s_i$ and $r_j$ belong to $Z_p$).

For each class $C_i$ with an ancestor set $S_i = \{C_{i_1}, C_{i_1}, \ldots, C_{i_n}\}$ where $i_j$ is an ordinal number such that $1 \le i_j \ne i \le n$, class $C_i$ is given a polynomial function, $g_i$ derived by the CA as,

$$g_i(x_{m_i+2}, x_{m_i+3}, \ldots, x_m) = P(s_i, s_{i_1}, s_{i_2}, \ldots, s_{i_m}, x_{m_i+2}, x_{m_i+3}, \ldots, x_m)$$

A symmetric polynomial based scheme:

$A$ is an set of n classes – $\{C_1, C_2, C_3, \ldots, C_n\}$

$B$ is a set of ancestral classes of set $A$.

$$B = \{S_1, S_2, S_3, \ldots, S_n\}$$

$mi$ is calculated as the number of the ancestral classes $m_i = |S_i|$. We choose m such that $m \ge \max\{m_1, m_2, \ldots, m_n\} + 1$. Here $m$ is the number of parameter in the polynomial function $P$, where $P$ is to construct our multi level access control scheme.

We illustrate with a sample hierarchy "**Figure 4**"

Here we have nine classes

$\{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\}$

Ancestral classes' sets are

$S_1 = \{\varphi\}$, $S_2 = \{\varphi\}$, $S_3 = \{C_1, C_2\}$, $S_4 = \{C_2\}$

$S_5 = \{C_2\}$, $S_6 = \{C_1, C_2, C_3\}$,

$S_7 = \{C_1, C_2, C_3, C_4\}$

$S_8 = \{C_2, C_3, C_5\}$, $S_9 = \{C_2, C_5\}$

From the previous step, we need to choose m such that $m \ge \max\{m_1, m_2, m_3, \ldots, m_9\}$. Let us choose $m = 7$, it will allow to expand the hierarchy without changing the value of $m$.

Symmetric polynomial, we are using here is as follows

$$P(x_1, x_2, \ldots x_m) = \sum_{i_1=0}^{t} \sum_{i_2=0}^{t} \cdots \sum_{i_m=0}^{t} a_{i_1, i_2, \ldots i_m}, \; x_1^{i_1}, x_2^{i_2}, \ldots x_m^{i_m} \bmod P$$

Here $t$ is threshold number. We can classify our work into two Key Calculation and Key Derivation.

Key Calculation:

We can calculate key $K_i$ of class $C_i$ as follows

$$K_i = P(s_i, s_{i_1}, s_{i_2}, \ldots, s_{i_m}, s_1', s_2', \ldots, s_{m-m_i-1}') \quad (1)$$

Key Derivation:

In key derivation, we are using a term $Sj/I$ which is

$$S_{j/I} = S_j / (S_i U \{C_i\}) = \{C_{(j/i)1}, C_{(j/i)2}, \ldots, C_{(j/i)rj}\}$$

Consider a class $C_i$ which is ancestor to class $C_j$ and key $K_j$ can be calculated by $C_i$ as,

$$K_j = g_i(s_j, s_{(j/i)1}, s_{(j/i)2}, \ldots, s_{(j/i)r_j}, s'_1, s'_2, \ldots, s'_{m-m_i-2-r_j}) = P(s_i, s_j, s_i, s_{i_1}, s_{i_2}, \ldots, s_{i_{m_i}}, s_{(j/i)1}, s_{(j/i)2}, \ldots, s_{(j/i)r_j}, s'_1, s'_2, \ldots, s'_{m-m_i-2-r_j}) \quad (2)$$

*Example Key Derivation*

Consider that $C_3$ is an ancestor class to class $C_7$.

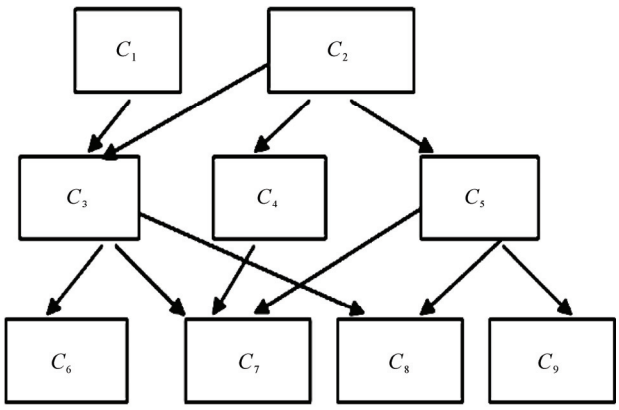


**Figure 4. A typical hierarchy.**

 

Then $K_7$ can be derived by $C_3$ in the following steps.

$$S_{7/3} = \{C_4\}$$
$$K_7 = P(s_3, s_7, s_1, s_2, s_4, s_1', s_2')$$

Key Calculation for the Classes using Equation (1)

$$K_1 = P(s_1, r_1, r_2, r_3, r_4, r_5, r_6)$$
$$K_2 = P(s_2, r_1, r_2, r_3, r_4, r_5, r_6)$$
$$K_3 = P(s_1, s_2, s_3, r_1, r_2, r_3, r_4)$$
$$K_4 = P(s_4, s_1, s_2, r_1, r_2, r_3, r_4)$$
$$K_5 = P(s_5, s_1, s_2, r_1, r_2, r_3, r_4)$$
$$K_6 = P(s_6, s_1, s_2, s_3, r_1, r_2, r_3)$$
$$K_7 = P(s_7, s_1, s_2, s_3, s_4, r_1, r_2)$$
$$K_8 = P(s_8, s_1, s_2, s_3, s_4, s_5, r_1)$$
$$K_9 = P(s_9, s_1, s_2, s_3, s_4, s_5, r_1)$$

Key Derivation of class 7 by class 3 using Equation (2)

$$S_3 = \{C_1, C_2\}$$
$$S_7 = \{C_1, C_2, C_3, C_4\}$$
$$S_3 U \{C_3\} = \{C_1, C_2, C_3\}$$
$$S_{3/7} = \{C_4\}$$
$$K_7 = P(s_3, s_7, s_1, s_2, s_4, r_1, r_2)$$

Which is equal to the key calculated by class7 itself.

Key Derivation of class 3 by class 7 using Equation (2)

$$S_3 = \{C_1, C_2\}$$
$$S_7 = \{C_1, C_2, C_3, C_4\}$$
$$S_7 U = \{C_7\} = \{C_1, C_2, C_3, C_4, C_7\} \quad (3)$$
$$S_{7/3} = \{\varphi\}$$
$$K_7 = P(s_7, s_3, s_1, s_2, s_3, s_4, s_1')$$

It can be seen that when the class derives its own key and when a ancestor of this class derives the key same parameters are passed in the polynomial but the combination differs when a wrong ancestor derives the key, the parameters are not the same.

The default values, we have taken are
$m = 7$, $P = 2147483646$, $s_1 = 5$, $s_2 = 10$, $s_3 = 13$, $s_4 = 9$, $s_5 = 6$, $s_6 = 22$, $s_7 = 18$, $s_8 = 30$, $s_9 = 39$, $r_1 = 11$, $r_2 = 12$, $r_3 = 13$, $r_4 = 14$, $r_5 = 15$, $r_6 = 16$, $r_7 = 17$, $r_8 = 18$, $r_9 = 19$ (instead of *s*' we have used *r*)

For a small Hierarchy "**Figure 5**", with more than two classes, we can easily illustrate our key calculations, where each class consists of several users.

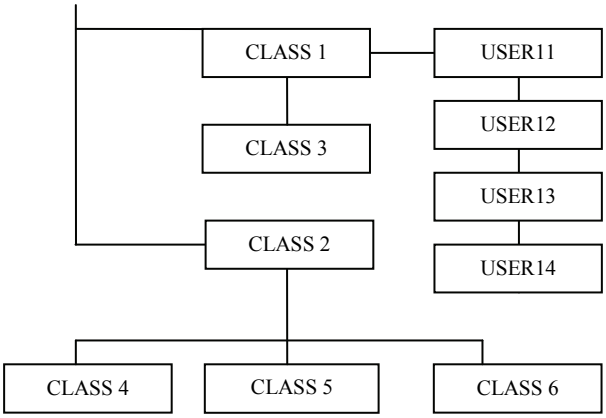


**Figure 5. Sample hierarchy to illustrate calculations.**

Key Calculations:
The parameters to be passed for calculating group key class 2 are $P(S_2, r_1, r_2, r_3, r_4, r_5, r_6)$
Group key class $C_2$ = 699615258
1) Private key for user 21 = 699615258 + (699615258/21) = 732930270
2) Private key for user 22 = 699615258 + (699615258/22) = 731415951
3) Private key for user 23 =699615258 + (699615258/23) = 730033312
4) Private key for user 24 = 699615258 + (699615258/24) = 728765893
Key Derivation:
Deriving the group key of class $C_4$ using its ancestral class $C_2$

$$S_2 = \{\ \}\quad S_4 = \{s_2\}$$

Sj|i can be calculated as
$S_{4|2} = \{s_4\} \big| \left(S_2 U \{C_2\}\right)$ Set Difference $S_{4|2} = \{\Phi\}$
The parameters to be passed for deriving the key of class $C_4$ using $C_2$

$$\text{Key} = p(s_2, s_4, r_1, r_2, r_3, r_4, r_5) = p(10,9,11,12,13,14,15)$$
$$= 1947982264$$

Private Keys are used for local communication.

## 6.3. Class Level Dynamics

### 6.3.1. Adding a Class

When a new class $C_r$ is added, we need to verify whether m value satisfies the new node constraints

1) If $m < \max\{m_1, m_2, ..., m_n, m_r\} + 1$, a new m value will be generated so that $m \geq \max\{m_1, m_2, ..., m_n, m_r\} + 1$. Also, the CA will regenerate a new polynomial functions $P(x_1, x_2, \ldots, x_m)$ accordingly. In addition, all polynomial functions of classes are recomputed and retransmitted securely.

 

2) If $m \geq \max\{m_1, m_2, ..., m_n, m_r\} + 1$, the CA selects a random number $s_r$ for the new class $C_r$ so that a new polynomial function $g_r$ can be computed and transmitted to class $C_r$ securely. However, if class $C_r$ is added as a parent class of any existing classes, we need to modify keys of $C_r$'s descendant classes to prevent class $C_r$ from obtaining old keys of its descendant.

#### 6.3.2. Deleting a Class

When a class $C_r$ is removed from the hierarchy, we need to determine whether the class $C_r$ is a leaf node or a parent node. Here, a leaf node is defined as a node without any descendant:

1) class $C_r$ is a leaf node: The CA can simply discard the public parameter $s_r$ without changing any other keys.

2) class $C_r$ is a parent node: Once class $C_r$ is deleted from the hierarchy, we cannot allow it to compute keys of $C_r$'s descendant classes using polynomial function $g_r$. We need to prevent class $C_r$ from accessing its descendants' resources.

#### 6.3.3. Moving a Class

A class $C_r$ can be moved from one node to another node in the hierarchy. There are four cases:

1) leaf node to another leaf node: the CA simply recomputes new polynomial function $g_r$ according the new hierarchy and securely transmits $g_r$ to $C_r$.

2) leaf node to parent node: the CA recomputes polynomial functions of class $C_r$ and $C_r$'s new descendant classes according to the new hierarchy. The CA securely transmits polynomial functions to the affected classes;

3) parent node to leaf node: the CA recomputes polynomial functions of previous descendant classes of $C_r$ and class $C_r$ according to the new hierarchy and then, securely transmits these polynomial functions to the affected classes

4) parent node to parent node: the CA recomputes polynomial functions of previous and present descendant classes of $C_r$ and class $C_r$ according to the new hierarchy and then, securely transmits these polynomial functions to the affected classes.

#### 6.3.4. Merging a Class

Two or more classes can merge together and become one class $C_r$. Similarly, the CA needs to find previous and present descendant classes of the merging classes. The CA randomly chooses a new number sr and then, generates polynomial functions for all corresponding classes.

#### 6.3.5. Splitting a Class

A class $C_r$ splits into two classes $C_{r1}$ and $C_{r2}$. Depending on whether $C_r$ is a parent node or leaf node, the CA has to determine what previous and present descendant classes are associated with these classes ($C_r$, $C_{r1}$ and $C_{r2}$). The CA then selects two new numbers $s_{j1}$ and $s_{j2}$ and generates polynomial functions for these affected classes.

#### 6.3.6. Adding a Link

If two classes $C_r$ and $C_k$ are linked together, we establish a new direct parent-child relationship between two classes, say class $C_r$ is the parent of class $C_k$. There are two different cases: 1) class $C_r$ was an ancestor of class $C_k$ through other classes. The CA does not need to perform anything; and 2) class $C_r$ is the only parent for class $C_k$ in the new hierarchy. The CA selects a new number $S_k$, and generates new polynomial functions for class $C_k$ and its descendants classes. The CA securely transmits new polynomial functions to these affected classes.

#### 6.3.7. Deleting a Link

If two linked classes $C_r$ and $C_k$ are disconnected, we destroy a direct parent-child relationship between two classes, say class $C_r$ will not be the parent of class $C_k$ in the new hierarchy. Again, there are two different cases: 1) class $C_r$ is still an ancestor of class $C_k$ through other classes in the new hierarchy. The CA does not need to perform anything; and 2) class $C_r$ is not an ancestor for class $C_k$ in the new hierarchy. The CA selects a new number $S_k$, and generates new polynomial functions for class $C_k$ and its descendants classes. The CA securely transmits new polynomial functions to these affected classes.

### 6.4. User Level Dynamics

In this scheme, every class represents certain access privileges. Also, a group of users in a class can share a key if they belong to the same class. For example, all users in class $C_j$ can compute the keys of class $C_j$ and its descendant classes. Dynamic user operations deal with how a user can join in a class or leave from a class, and possible displacement from one class to a different class. They all require the class key to be changed after any user operation is completed so that the issue of backward secrecy and forward secrecy can be addressed. Specifically, our scheme can revoke a user from a class $C_j$. It is as quick and efficient as to join a user in the class $C_j$. Both operations require that the CA randomly select a new public parameter $s_j$ for $C_j$ and recompute a new polynomial function $g_j$ by using the new $s_j$. Since the polynomial function $g_j$ is newly produced, other polynomial functions and keys are also recomputed for the descendant classes of $C_j$. This will guarantee both backward secrecy and forward secrecy. The efficiency can be improved if backward secrecy or forward secrecy is not required. Another common user operation is to allow a

 

user to move from one class $C_j$ to another class $C_k$. Here, the CA will randomly choose two new public parameters $s_j$ and sk for $C_j$ and $C_k$ so that new polynomial functions and keys are recomputed and transmitted to $C_j$, $C_k$ and their descendants respectively. Thus, both backward secrecy and forward secrecy are guaranteed.

#### 6.4.1. User Join

Every time if a single user wants to join a group the CA just allows the user to be added to the hierarchy and generates a private for that user by providing the corresponding group key. When a new user joins the hierarchy, it should be provided with a group key and there are no changes to be made on the user's key.

#### 6.4.2. User Leave

When a user wants to leave from the hierarchy the CA change the group key by making changes on anyone of the following changing the polynomial or changing the value factor $P$.

## 7. Simulation Results

The system is developed using .NET and found to be secure and fast. The system takes care of USER level and class level dynamics. The large number of numbers prevents a possible guessing. e.g., for a eight parameter polynomial, 16! (*i.e.*, 10922789888000) combinations possible. Bursty leave and join operations also are possible and the system can be used for any hierarchy. The outputs are shown in **Figures 6-10**.

## 8. Performance Analysis

Performance and security: Each user $u_i$ will receive

$$g_i = f(s_i, x_2, \ldots x_n) = g(x_2, \ldots x_n) = \sum_{i_2=0}^{i_2=w} \cdots \sum_{i_n=0}^{i_n=w} a_{i_2,\ldots i_n}, x_2^{i_2}, \ldots x_n^{i_n}$$

The time complexity for computing the group key is $O(w^{2n})$. An important measure for a secure group com munication scheme is the number of rekeying messages. Suppose that t users will be joining the group. The TA will send $k$ and $g_i$ to each of them respectively ($2t$ messages) and broadcast one message to tell which users are joining. The total number of rekeying messages is $O(2t)$. Suppose that $t$ users are leaving the group. The TA only broadcasts one message to tell which users are leaving, thus the number of rekeying messages is $O(1)$. Suppose that t users are joining and another $v$ users are leaving the group, the total number of rekeying messages is still $O(2t)$.

As for the security of the scheme, if $w + 1$ class collude, then they can Figure out the function f entirely. Therefore, the scheme is $w$-resilient. Moreover, if less than $w + 1$ classes collude, they cannot get any information about the key, *i.e.*, any value in the key space looks like a valid and equiprobable key to these colluding users. It follows that the scheme is unconditionally secure.

### 8.1. Memory

Each user will be able to calculate the key based on the

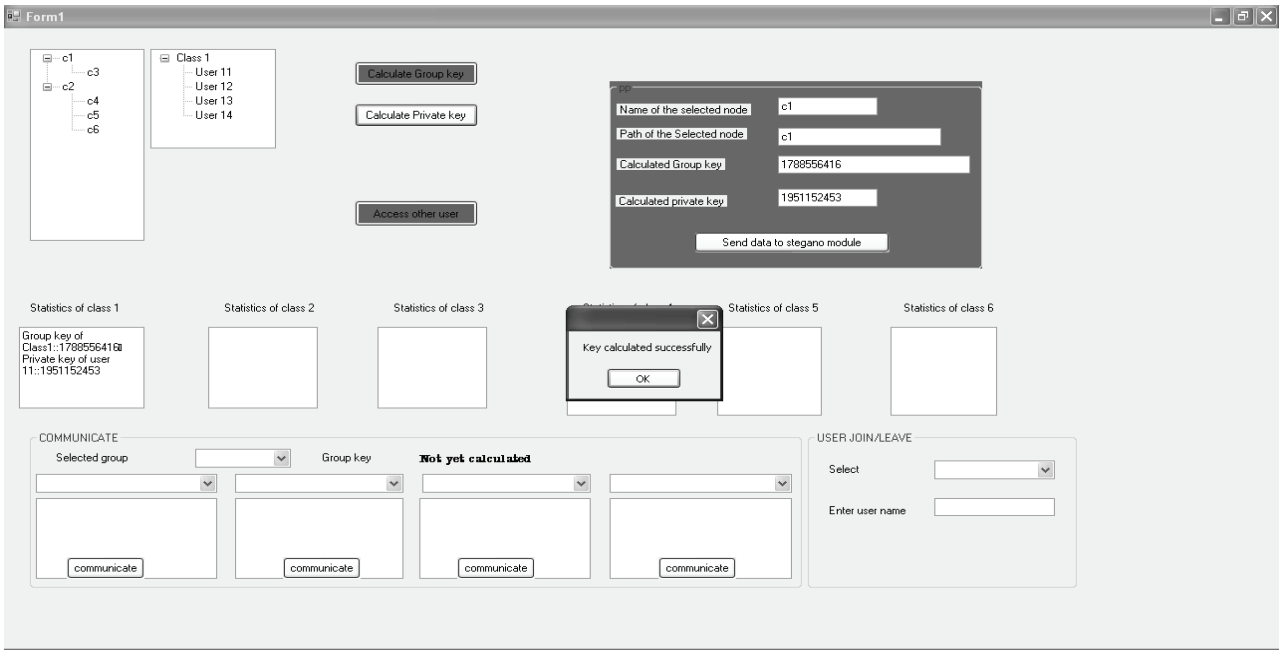


**Figure 6. Users and classes.**

 

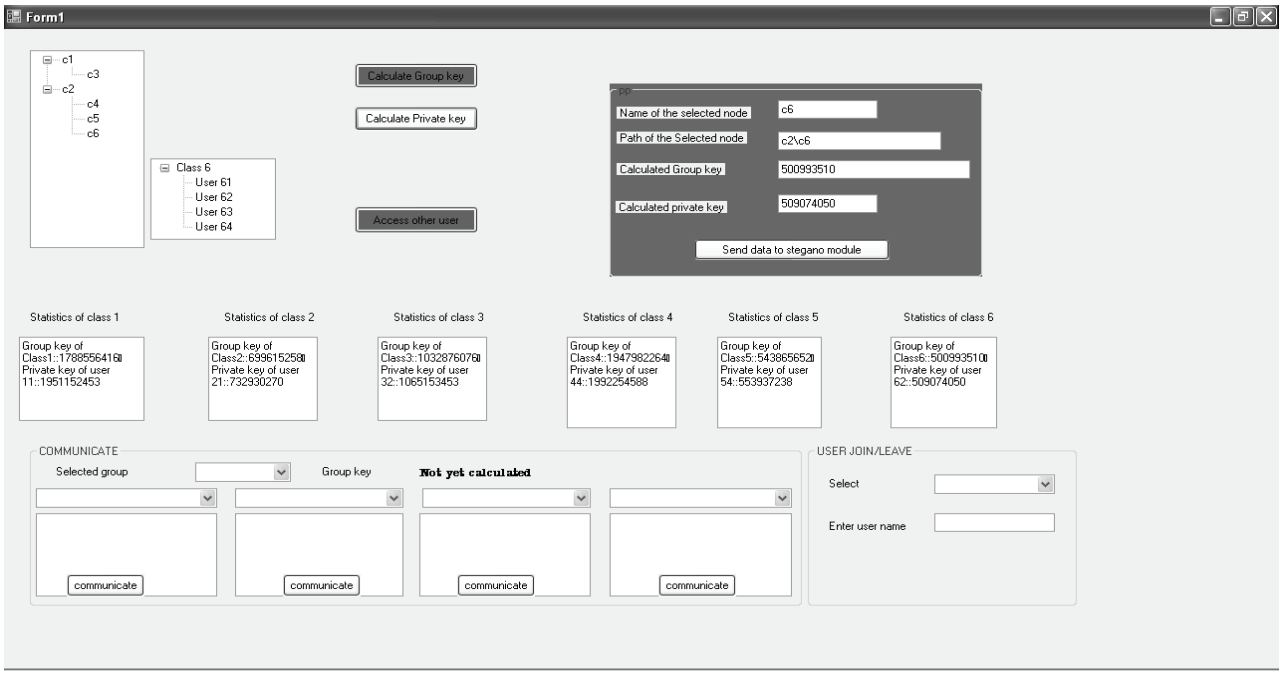


**Figure 7. Key calculation for all classes.**

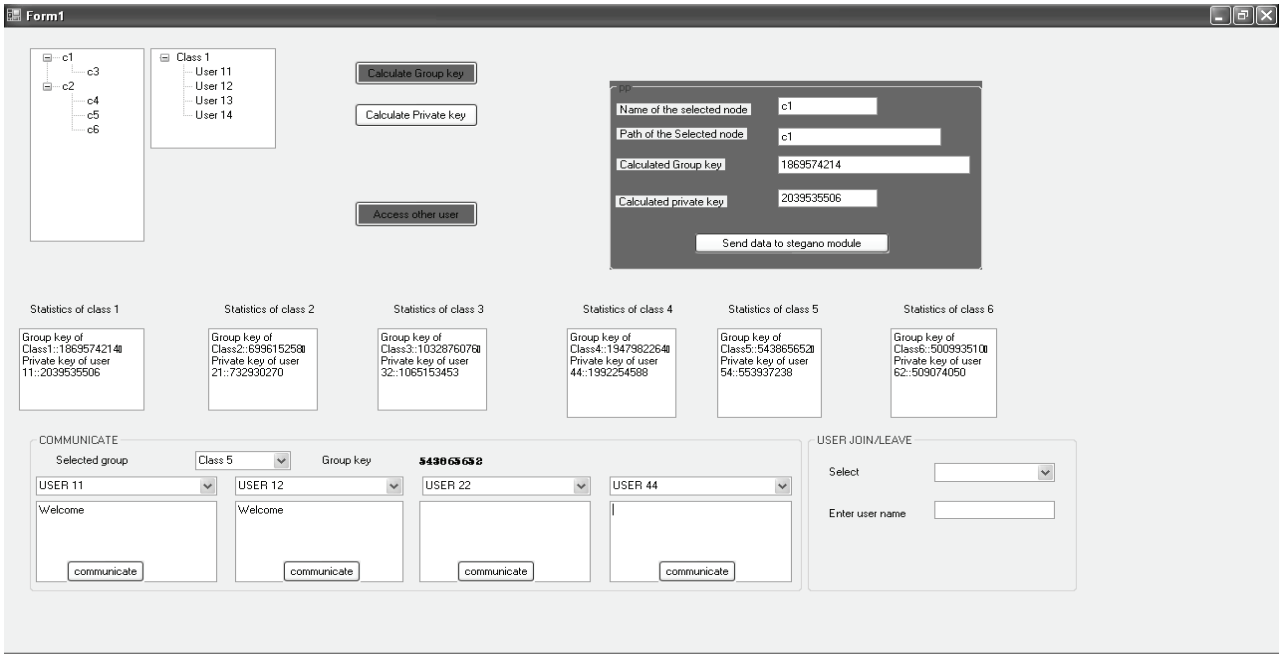


**Figure 8. Communication among same class.**

polynomial and hence very less memory is used. All parameters are publicly available and using the same method keys of lower hierarchy can be derived by substituting the corresponding parameters as given by the derivation module. The steganography module does not involve any storage for storing the already recorded data as always data is recorded and subsequently sent to the receivers. The size of the cover medium does not increase because only two bits are used in each channel.

## 8.2. Computation Cost

When the user joins there is no need for recalculation because the recorded message has already been played. When a user leaves a group key is recalculated and given to the class. Private keys are generated from this. The

 

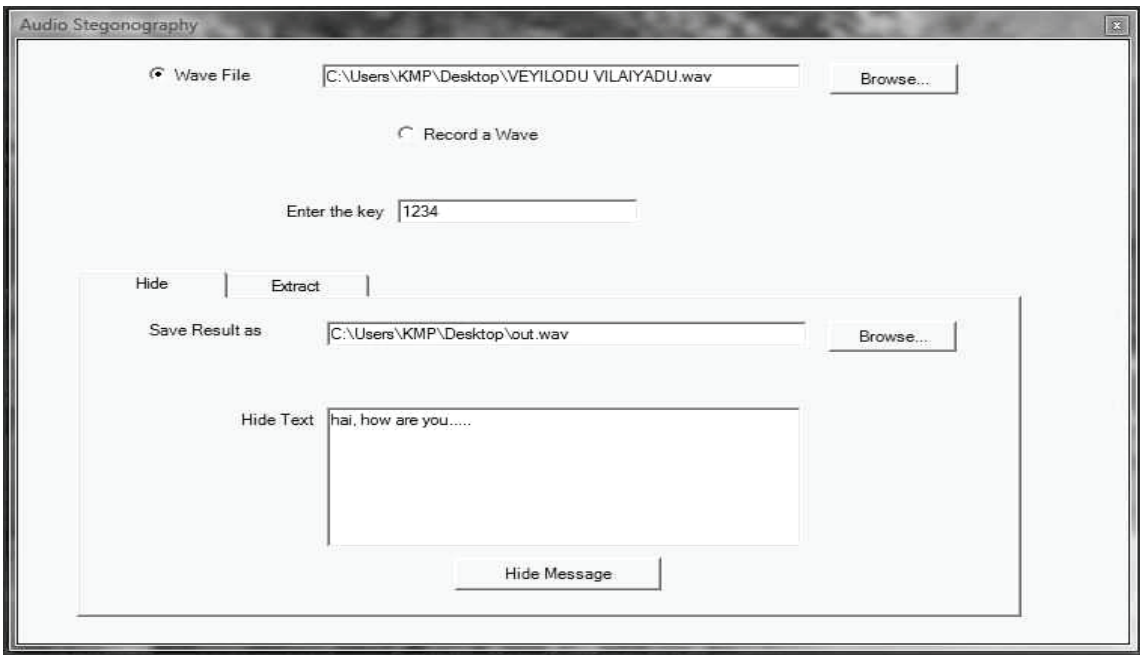


**Figure 9. Hiding the message.**

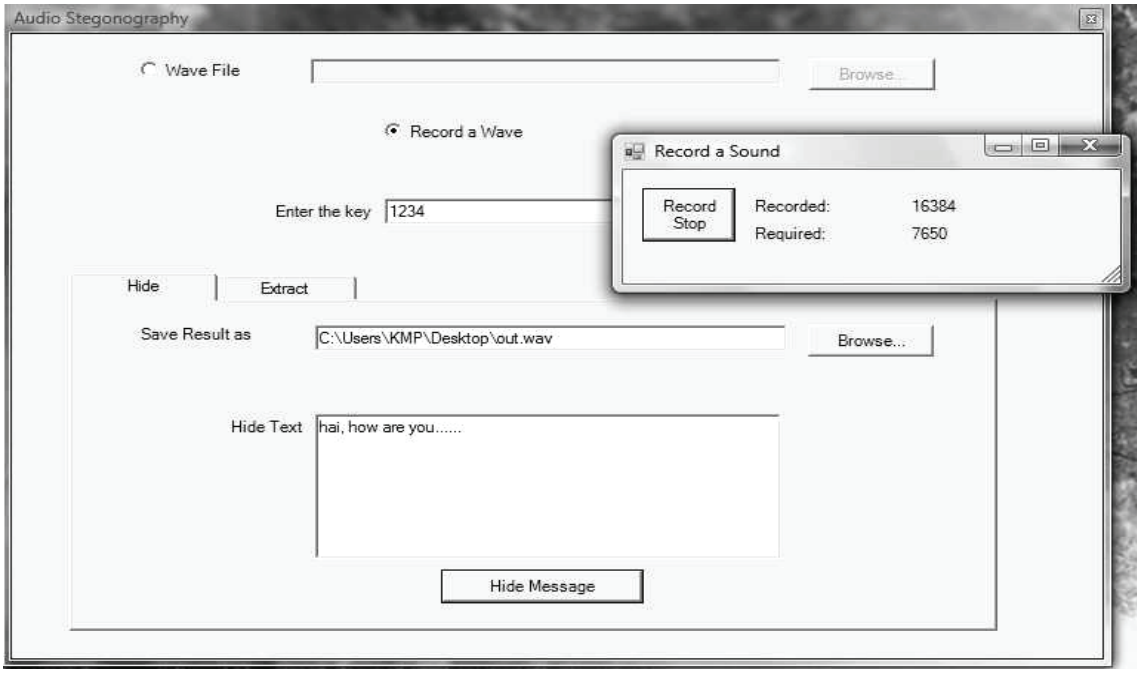


**Figure 10. Recording the message.**

value of the new key involves not a change of parameters but a change of mod *P* value. Hence each class will be able to get a new polynomial value by passing the same parameters. Any left user will not be able to get the key. Only one key is used during creation of stego data. The higher class users need not remember the keys of all their descendant classes but rather using a simple scheme derive the exact parameters to generate the key. Hence the computation cost is reduced.

## 8.3. Communication Cost

The P value is changed by the Trusted Authority and when the users try to calculate the key the new key will be generated. The computation cost is reduced because, the class users are not bothered about the key transmis-

 

sion. Once the polynomial is given the users can calculate their own key.

### 8.4. Dynamics

Class joining, Class leaving, Dynamic Hierarchy and New user joining a class are all done by the trusted authority in a phased manner thereby allowing the scheme to scale to greater hierarchy. Additionally local messaging and service messages are taken care in this system.

### 8.5. No of Rekeys

To calculate the no of changes might be made on the user's key based on user join/leave. Key must be changed when a new user is being joined/left from the hierarchy In the Hierarchy "**Figure 11**" for analyzing the no. of rekeys there are 50 classes and Let every classes consists of four users each, then we can calculate an unique key for every class based on this group key, we can generate a private key for every single user.

Classes:

Totally fifty classes of eight levels

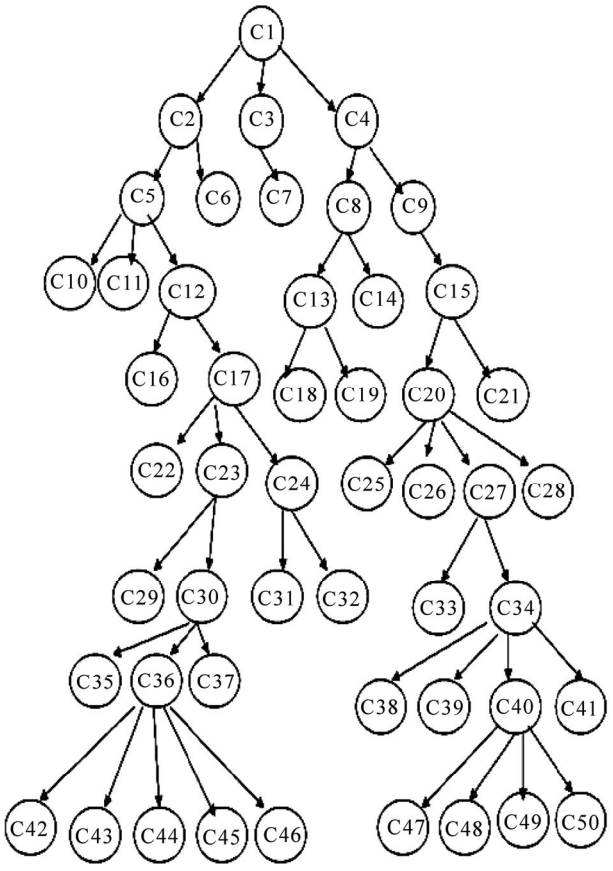


**Figure 11. Hierarchy analyzed for calculating the no. of rekeys.**

Users:

Every class is consists of 4 users

User joins:

Every time if a single user wants to join a group the CA just allows the user to be added to the hierarchy and generates a private for that user by providing the corresponding group key. The secrecy is maintained as the audio is real time.

User leave:

When a user wants to leave from the hierarchy the CA change the group key by making changes on anyone of the following Changing the polynomial, or by Changing the value factor P. The total no of key changes to be made = No. of Ancestor classes + 1 as shown in **Table 1**.

## 9. Conclusions and Future Work

### 9.1. Conclusions

Thus in this Paper, Design And Implementation Of Multilevel Access Control In Synchronized Audio To Audio Steganography Using Symmetric Polynomial Scheme is implemented successfully .The implementation results show that any type of hierarchy can be introduced and all dynamics can be done. For a 8 parameter polynomial that can have use 8 among the 16 values there are 16! combinations. Also, nodes can derive their descendants' keys without involvement of the CA once polynomial functions were distributed to them. In addition, the storage requirement and computation complexity at the CA are almost same as that at individual nodes, thus, the CA would not be a performance bottleneck and can deal with dynamic operations efficiently.

**Table 1. No of rekeys for user leaving from a corresponding class.**

| | | | | |
|---|---|---|---|---|
| Class 1: 1 | Class 11:4 | Class 21:5 | Class 31:7 | Class 41:8 |
| Class 2:2 | Class 12:4 | Class 22:6 | Class 32:7 | Class 42:9 |
| Class 3:2 | Class 13:4 | Class 23:6 | Class 33:7 | Class 43:9 |
| Class 4:2 | Class 14:4 | Class 24:6 | Class 34:7 | Class 44:9 |
| Class 5:3 | Class 15:4 | Class 25:6 | Class 35:8 | Class 45:9 |
| Class 6:3 | Class 16:5 | Class 26:6 | Class 36:8 | Class 46:9 |
| Class 7:3 | Class 17:5 | Class 27:6 | Class 37:8 | Class 47:9 |
| Class 8:3 | Class 18:5 | Class 28:6 | Class 38:8 | Class 48:9 |
| Class 9:3 | Class 19:5 | Class 29:7 | Class 39:8 | Class 49:9 |
| Class 10:4 | Class 20:5 | Class 30:7 | Class 40:8 | Class 50:9 |

 

### 9.2. Future Work

1) The Trusted authority can still made secure by changing the value of the parameters.

2) A symmetric polynomial can be changed by the trusted authority.

3) Bit selection for steganography can be made by using some pseudo random generator.

## 10. References


[1] W. Stallings, Ed., "Network and Internetworking Security," Pearson Education Asia, Singapore, 2001.

[2] N. F. Johnson, Z. Duric and S. Jajodia, "Information Hiding Steganography and Watermarking-Attacks and Countermeasures," Kluwer Academic Publishers, Boston, 2001.

[3] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding - A Survey," *Proceedings of IEEE*, Vol. 87, No. 7, 1999, pp. 1062-1078.

[4] M. Hosei, "Acoustic Data Hiding Method Using Sub-Band Phase Shifting," *Technical Report of IEICE, EA*, Vol. 106, No. 205, 2006, pp. 7-11.

[5] M. Wu and B. D. Liu, "Multimedia Data Hiding," Springer-Verlag, New York, 2003.

[6] T. Aoki and N. Homma, "A Band Widening Technique for VoIP Speech Using Steganography Technology," *Report of IEICE*, *SP*, Vol. 106, No. 333, 2006, pp. 31-36.

[7] X. P. Huang, R. Kawashima, N. Segawa and Y. Abe, "The Real-Time Steganography Based on Audio-to-Audio Data Bit Stream," *Technical Report of IEICE, ISEC*, Vol. 106, No. 235, September 2006, pp. 15-22.

[8] X. P. Huang, R. Kawashima, N. Segawa and Y. Abe, "Design and Implementation of Synchronized Audio-to-Audio Steganography Scheme," *IEEE Explore*, 2008, pp. 331-334.

[9] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," *ACM Transactions on Computer Systems*, Vol. 1, No. 3, March 1983, pp. 239-247.

[10] S. J. MacKinnon, P. D. Taylor, H. Meijer and S. G. Akl, "An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy," *IEEE Transactions on Computers*, Vol. 34, No. 9, September 1985, pp. 797-802.

[11] S. Chen, Y.-F. Chung and C.-S. Tian, "A Novel Key Management Scheme for Dynamic Access Control in a User Hierarchy," *COMPSAC*, September 2004, pp. 396-397.

[12] I. Ray, I. Ray and N. Narasimhamurthi, "A Cryptographic Solution to Implement Access Control in a Hierarchy and More," *SACMAT*'02: *Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*, ACM Press, New York, 2002, pp. 65-73.

[13] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," *Information Processing Letter*, Vol. 27, Vol. 2, February 1988, pp. 95-98.

[14] G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," *Proceedings on Advances in Cryptology*: *CRYPTO*'89, *LNCS*, Vol. 435, 1989, pp. 316-322.

[15] M. L. Das, A. Saxena, V. P. Gulati and D. B. Phatak, "Hierarchical Key Management Scheme Using Polynomial Interpolation," *SIGOPS Operating Systems Review*, Vol. 39, No. 1, January 2005, pp. 40-47.

[16] L. Harn and H. Y. Lin, "A Cryptographic Key Generation Scheme for Multilevel Data Security," *Computers and Security*, Vol. 9, No. 6, October 1990, pp. 539-546.

[17] V. R. L. Shen and T.-S. Chen, "A Novel Key Management Scheme Based on Discrete Logarithms and Polynomial Interpolations," *Computers and Security*, Vol. 21, No. 2, March 2002, pp. 164-171.

[18] M.-S. Hwang, C.-H. Liu and J.-W. Lo, "An Efficient Key Assignment for Access Control in Large Partially Ordered Hierarchy," *Journal of Systems and Software*, February 2004.

[19] C. H. Lin, "Dynamic Key Management Scheme for Access Control in a Hierarchy," *Computer Communications*, Vol. 20, No. 15, December 1997, pp. 1381-1385.

[20] S. Zhong, "A Practical Key Management Scheme for Access Control in a User Hierarchy," *Computers and Security*, Vol. 21, No. 8, November 2002, pp. 750-759.

[21] X. Zou, B. Ramamurthy and S. Magliveras, "Chinese Remainder Theorem Based Hierarchical Access Control for Secure Group Communications," *Lecture Notes in Computer Science* (*LNCS*), Vol. 2229, November 2001, pp. 381-385.

[22] X. Zou, B. Ramamurthy and S. S. Magliveras, Eds., "Secure Group Communications over Data Networks," Springer, New York, October 2004.


 