

# Multilevel Secure Teleconferencing Over Public Switched Telephone Network

Inja Youn<sup>1</sup>, Csilla Farkas<sup>2</sup>, and Bhavani Thuraisingham<sup>3</sup>

<sup>1</sup> Department of Information and Software Engineering,  
George Mason University, Fairfax, VA 22030.  
`iyoun@gmu.edu`

<sup>2</sup> Dept. of Computer Science and Engineering,  
University of South Carolina, Columbia, SC 29208.  
`farkas@cse.sc.edu`

<sup>3</sup> Erik Jonsson School of Engineering and Computer Science,  
University of Texas at Dallas, Richardson, TX 75080.  
`bhavani.thuraisingham@utdallas.edu`

**Abstract.** Two-way group voice communications, otherwise known as teleconferencing are common in commercial and defense networks. One of the main features of military teleconferences is the need to provide means to enforce the Multilevel Security (MLS) model. In this paper we propose an architecture and protocols facilitating MLS conferences over Public Switched Telephone Network (PSTN). We develop protocols to establish secure telephone conferencing at a specific security level, add and drop conference participants, change the security level of an ongoing conference, and tear down a conference. These protocols enforce MLS requirements and prevent against eavesdropping. Our solution is based on encryption methods used for user and telephone authentication and message encryption, and trusted authentication centers and certificate authorities. We provide an initial estimate of signaling delays of our protocols incurred due to the enforcement of the MLS requirements.

## 1 Introduction

The need to provide secure communication via public telephone systems has resulted in custom designed and dedicated devices, like the secure telephone unit third generation (STU-III) [3] and TeleVPN [2]. While these methods provide some level of confidentiality, they require extensive setup procedures and dedicated hardware or do not require telephone device authentication. Our aim is to enable current telephone technologies to provide voice privacy without the extensive setup and maintenance requirements of the current systems.

Public Switched Telephone Networks (PSTN [13] - a circuit switched network with almost zero down time and acceptable quality audio signals - use Signaling System 7 (SS7) [4, 5, 9, 7, 8, 6, 11] as its signaling network to set up, configure, maintain, and tear down voice circuits that are used to transmit continuous voice

streams. Moreover, increasingly popular mobile telephones can also depend on SS7. However, SS7 provides limited security to its signaling and voice networks, as shown by Lorencz et al. [10]. Recognizing these limitations, Sharif et al. [12] present protocols to ensure voice confidentiality over PSTN using the Discretionary Access Control (DAC) model. Their solution uses public and secret key encryption methods to authenticate the users and telephone devices, and to provide encrypted end-to-end communication. They show that authentication delays are within acceptable range for PSTN. Youn et al. [14] extend the protocols of Sharif et al. to DAC based secure teleconferences over PSTN. That is, participation in a conference is decided on the identity of the user (telephone device). However, their methods do not satisfy the security needs of military conferences. In this paper, we extend both these works to MLS based teleconferencing. We adopt the Bell-LaPadula (BLP) [1] access control model.

BLP policies are expressed via security classification labels, assigned to subjects (i.e., active computer system entities) and to objects (i.e., passive resources). Classification labels form a lattice with a dominance relation among the labels. BLP controls read and write operations on the objects based on the classification labels of the requested data objects and the clearance of the subject requesting the operation. For example, BLP ensures that a subject can read an object only if the subject's clearance dominates the object's classification (simple-security property) and that a subject can write an object only if the object's classification dominates the subject's clearance (\*-property). Trusted subjects are permitted to bypass the \*-property of the BLP. The two axioms of BLP ensure confidentiality by permitting information flow from a dominated security class to a dominating security class but not in the other direction. While MLS is considered too restrictive for general purpose applications, it is required in the military domain.

In this paper we propose an MLS teleconference security model and provide a set of protocols to establish and maintain an MLS teleconference at a specified security level. In our model, a user (conference participant) and his/her telephone device together are considered as the subject; the conference (i.e., its content) is considered as the object. The user who initiates the conference, called call controller, requests the join (add) of a user/telephone to an active conference. However, the actual "adding" of a user/telephone must be permitted by a referential monitor that enforces the simple security property of BLP. That is, a user/telephone is permitted to join a conference only if the security classification of the conference is dominated by the greatest lower bound of the security clearances of the user and the telephone device. The human users are trusted not to violate the \*-property, i.e., a user is trusted not to reveal any information that is classified higher than the level of the conference. Call controllers are also trusted (trusted subject) to lower the security clearance of an ongoing conference. To ensure that telephone devices cannot leak confidential information, they are cleared based on their encryption capabilities and verified hardware. We develop a set of protocols to ensure that the conference content is protected from unauthorized disclosure at any time. We also perform analysis

of the conference dynamics and the necessary security evaluations to guarantee message confidentiality. Our aim is to limit the necessary delays incurred by the authentication, security checking, and the conference key refreshment. We give an analysis of the incurred delays for our secure teleconference.

The organization of the paper is as follows. Section 2 introduces our security architecture and the MLS teleconferencing model. In Section 3 we present descriptions of our protocols and the corresponding security requirements. Section 4 contains the delay calculation. Finally, we conclude and recommend future research directions in Section 5. We included sample protocols in Appendix A and the break down of the delay calculation in Appendix B.

## 2 Security Model

The main aim of our research is to build on top of the existing communication infra-structure. Our protocols to set up, maintain, and tear down secure teleconferences use libraries on the Signaling System 7 (SS7) protocol stack. MLS teleconferencing uses secure bridges [12, 14].

### 2.1 Secure Teleconferencing Architecture

We distinguish between a single master-secure bridge (MSB) and slave-secure bridges (SB). MSB has all the capabilities needed for teleconferencing and to enforce MLS requirements. MSB connects to the call-master, i.e., the participant who is initiating the conference. Slave-secure bridges (SB), connecting the conference participants, perform participant and telephone authentication. Each secure bridge is associated with an 1) Authentication Center (AC) to authenticate users and telephones, and to manage secret keys, and a 2) Certificate Authority (CA) to manage digital certificates and generate public/private key pairs. Our model requires that each telephone has cryptographic capabilities using symmetric and public keys. Telephones (and their corresponding secure bridges) are trusted based on these cryptographic capabilities as well as hardware verification of the physical device.

Additional PSTN components, like the Service Switching Points (SSP), Service Control Points (SCP), and Signal Control Point, together with the secure bridges form the secure teleconferencing architecture [14]. Our protocols use the Digital Subscriber Signaling System no 1 (DSS1) to communicate between the telephones and the local SSPs. ISDN user part (ISUP) is used for communication between SSPs and Transaction capabilities Library (TCAP) as well as for transactions between SSPs, ACs, CAs, and Line Information Translation Database (LIDBs).

### 2.2 Security Model

Our goal is to protect the confidentiality of the telephone conversation from unauthorized disclosure. Note, that the problem of hiding the existence of an

unauthorized conference is outside of the scope of this paper. We propose methods to apply the BLP security model to teleconferencing. The subject of our model is the telephone device and the human user (conference initiator and participants) using the telephone. Telephones are authenticated based on the telephone line numbers (TLN), telephone device numbers (TDN), and the private keys assigned to them. A security clearance label is assigned for each telephone, based on its encryption capabilities, verification of hardware components (e.g., trusted hardware and reliability), and physical security. Telephone clearances are considered relatively static. Increasing or decreasing a telephone's clearance level requires technical modifications, like encryption updates. We assume that users are aware of the clearance of the telephone devices.

User authentication is performed by a claimed user identity and the corresponding password. Each user with maximum security clearance  $\lambda$  is associated with a set of passwords, where each password  $\lambda'$  in the set corresponds to a specific security level and  $\lambda \geq \lambda'$ . To prevent the exposure of a higher security password on a lower security telephone device, we require that each user is authenticated with the password that is assigned to him/her for the level of the telephone device. For example, a user  $U$  with Top-Secret (TS) security clearance has different passwords for Unclassified, Secret, and Top-Secret levels. When  $U$  uses a telephone with Secret clearance, the user is authenticated based on his/her Secret level password. Note, that different approaches could be used to limit exposure of user passwords on telephones. For example, users may be restricted to use telephone devices only if the clearance level of the device dominates the clearance level of the user. Finding the optimal approach is outside of our current research and is dependent on the application area, the number of levels, and the available hardware resources.

A secure bridge, serving a telephone with clearance  $\lambda$ , stores the appropriate (user-id, password) pairs for all levels  $\lambda'$ , where  $\lambda \geq \lambda'$ . For each call activation by a user  $U_i$ , using the telephone  $T_i$ , the permitted security clearance is calculated as the greatest lower bound of  $[\lambda(U_i), \lambda(T_i)]$ , where  $\lambda(U_i)$  and  $\lambda(T_i)$  are the clearances of user  $U_i$  and device  $T_i$ , respectively.

The protection object is the content of the telephone conference. Each conference is initiated at a specified security level. Conference classification levels may increase and decrease along the dominance relation of the security lattice. We require that a user/telephone pair is permitted to initiate or join a conference only if the greatest lower bound of their joint security clearance dominates the security classification of the conference.

This paper studies the conference dynamics, including initiating the conference, adding and dropping participants, changing security classification of an ongoing conference, and changing the call controller of an ongoing conference. Our security requirement is that an unauthorized user should not be able to disclose the conference content. That is, unauthorized users should not be permitted to become participants of a conference or gain access to the secret key used to encrypt the content of the conference. The later requirement protects against passive eavesdropping. Our security requirements are supported by the proper-

ties of existing secret and public key encryption methods and by safeguarding the encryption/decryption keys. In addition to the security requirement we want to limit the number of authentication procedures and key updates that cause delays in the teleconferencing.

### 3 Protocols

We developed eight protocols to support secure telephone conferencing: 1) Establish a conference, 2) Add a new conferee by the call controller, 3) Add a new conferee by his/her own request, 4) Drop a conferee by the call controller, 5) Drop a conferee by his/her own choosing (hang up), 6) Change the classification of an ongoing conference 7) Call teardown by the call controller hanging up, and 8) Call teardown when the last slave conferee hangs up. Due to the space restrictions of the paper we only present some of our protocols.

#### 3.1 Protocol 1 - Teleconference Call Setup Process

The teleconference call setup process has five phases: 1) Telephone authentication, 2) User authentication, 3) Cross certification of the MSB, 4) Remote telephone authentication, 5) Remote user authentication, 6) Cross certification of the SSBs, and 7) Key distribution. Figure 1 shows the control messages for conference set up steps and the authentication of the call controller. The protocol is given below. The call controller's (user  $U_0$ ) security clearance is determined by the security clearance of the telephone device ( $T_0$ ) used to initiate the conference and the security clearance of the  $(U_0, password_0)$  pair. The permitted classification level of the conference to be initiated by  $U_0$  is  $\lambda(U_{0-permitted}) = GLB[\lambda(T_0), (\lambda(U_0, password_0))]$ . The call controller is permitted to initiate a conference with security classification  $\lambda'$ , where  $\lambda(U_{0-permitted}) \geq \lambda'$ . The actual protocol steps are given in Appendix A.

1. [ $T_0$ ] The call controller ( $U_0$ ) dials the teleconference access code. Once the telephone enters the teleconference mode, the call controller enters the telephone line number (TLN) of the master secure bridge (MSB).
2. [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ]  $T_0$  invokes the facility that initiates the conference sending  $M_1 = K_{msb}^*[K_0^*[TLN_0, TDN_0, t_0]]$  to  $MSB$ , where  $K_{msb}$  is the public key of the  $MSB$ , and  $K_0^*$  is the private key of  $T_0$ .

This message is used for the authentication of the telephone device and travels in the SETUP message (ISDN) between  $T_0$  and  $SSP_0$ , and in the IAM primitive (ISDN) between  $SSP_0$  and  $SSP_{msb}$ . While the IAM message travels through the SS7 network, the intermediate exchanges allocate the voice trunks. The destination exchange ( $SSP_{msb}$ ) allocates the resources for the secure teleconference (the Master Secure Bridge -  $MSB$ ) and initiates the teleconference transaction by sending the message  $M_1$  to  $AC_{msb}$ .

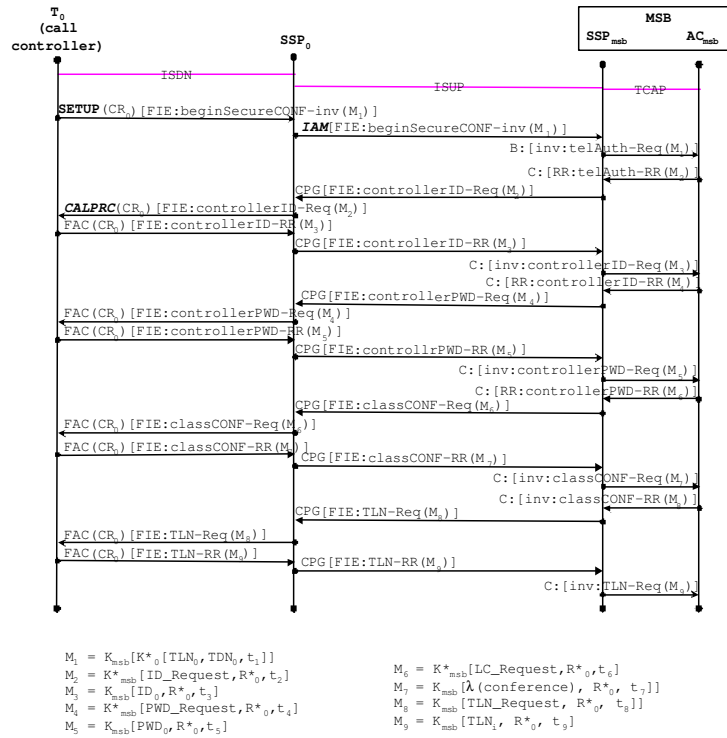


Fig. 1. Conference Call Setup: Initiation and call controller authentication

3.  $[AC_{msb}]$  The authentication center of the Master Secure Bridge verifies the authenticity of the telephone set by extracting  $TLN_0$  and  $TDN_0$  and comparing them against the ones stored in the database. It also checks the validity of the timestamp to prevent the replay attack. The authentication center looks in its database for the telephone classification  $\lambda(T_0)$ .

**If** authentication succeeds and the algorithm continues with the step 4.

**Else**  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair that propagates along the allocated path.

4.  $[AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0]$   $MSB$  replies with a request for user authentication embedded in a Call Progress (CPG) message:

$M_2 = K_{br}^*[ID\_Request, R^*_0, t_1]$ , where  $R^*_0$  is a nonce generated by  $AC$  that will be embedded in the message exchanged between call master and  $MSB$  during the teleconference session, and  $t_1$  is a timestamp. Both the random

number and the timestamp are meant to prevent the replay attack. An IVR message solicits the user to dial her user ID.

5.  $[T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}]$  The call controller enters her/his ID ( $ID_0$ ):  $M_3 = K_{msb}[ID_0, R_0^*, t_3]$
6.  $[AC_{msb}]$  The authentication center of the *MSB* decrypts  $M_3$  and checks the validity of the random number, timestamp, and looks in the database for  $ID_0$ .

If authentication succeeds the protocol continues with the step 7.

**Else**,  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair and ends the transaction with  $AC_{msb}$

7.  $[AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0]$  The authentication center sends a signed acknowledgement in a CPG message, which contain a request for password:  $M_4 = K_{msb}^*[PWD\_Request, R_0^*, t_4]$
8.  $[T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}]$  The call controller dials her password ( $PWD_0$ ), which will be again send to  $AC_{msb}$  in a CPG message encrypted by the public key of *MSB*.  $M_5 = K_{msb}[PWD_0, R_0^*, t_5]$
9.  $[AC_{msb}]$  The *MSB* decrypts the message and checks the timestamp and the  $(ID_0, PWD_0)$  pair.

If authentication succeeds, i.e., there is an  $(ID_0, PWD_0)$  pair, the  $AC_{msb}$  maps the user clearance  $\lambda(U_0, password_{U_0})$ .  $AC_{msb}$  computes  $\lambda(permitted) = GLB[\lambda(T_i), \lambda(U_0, password_{U_0})]$ . The protocol continues to assign security classification for the conference.

**Else**,  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair and ends the transaction with  $AC_{msb}$

### 3.2 Protocol 2 - Add a Participant to an Ongoing Conference

After the conference is set up, new participants  $U_x$  may join the ongoing conference.  $U_0$  (call controller) places the teleconference on hold by pressing the HOLD button. The other conferees are still able to talk while the conference is on hold.  $U_0$  initiates the new participant by dialing the  $U_x$ 's telephone number. The minimal requirement after successful authentication of  $T_x$  and  $U_x$  is that  $GLB[\lambda(T_x), \lambda(U_x, password_x)] \geq \lambda(conf)$ . Based on the conference dynamics, the encryption key used for the conference may or may not need to be updated (see Section 3.4).

### 3.3 Protocol 3 - Drop a Participant from an Ongoing Conference

Conference participants may be dropped from an active conference voluntarily (conferee hangs up) or non-voluntarily (call controller drops the user to maintain the MLS requirements). For example, a user with Secret clearance may decide to discontinue participation in a Secret conference. The same user may rejoin the conference at a later time. On the other hand, a user with Secret clearance is "forced" to be dropped from a conference when the conference classification is increased from Secret to Top-Secret. The *MSB* is responsible for enforcing the drop of the participants, reallocating the system resources, and initiating a new encryption key if a forced drop occurred.

### 3.4 Protocol 4 - Change the Security Classification of an Ongoing Conference

The security classification of an ongoing conference may be changed during the conference. For example, after discussing a Top-Secret topic, the security classification of the conference may be decreased to Secret to allow participation of Secret users. Any change in the conference classification may have an effect on the 1) minimum clearance requirement of the call controller, 2) new clearance requirements of the participants of the ongoing conference, 3) dropping conference participants, and 4) need of new encryption key. Figure 2 shows the message transfer to change the conference classification.

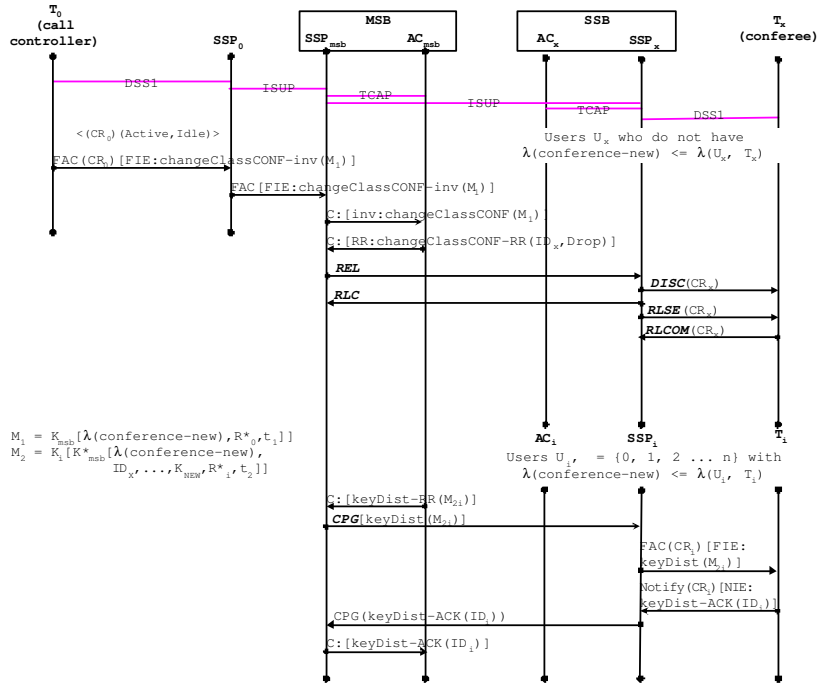


Fig. 2. Changing an ongoing conference classification

To change the security classification of an ongoing conference to a new classification, the call controller  $U_0$  must be cleared to the new classification. That

is, if  $\lambda(\text{conf}_0)$  denotes the security classification of the ongoing conference, and  $\lambda(\text{conf}_{\text{new}})$  denotes the requested security classification, then the new level is permitted only if  $\lambda(U_0) \geq \lambda(\text{conf}_{\text{new}})$ . Moreover, to decrease the classification of a conference, the call controller must be trusted. If  $\lambda(U_i) \geq \lambda(\text{conf}_{\text{new}})$  is not true for all participants  $U_i$  then  $U_i$  must be dropped and a new message encryption key must be distributed among the remaining participants. Also, if the conference classification is decreased and a new user  $U_i$  is added such that  $\lambda(\text{conf}_0) > \lambda(U_i) \geq \lambda(\text{conf}_{\text{new}})$  then a new message encryption key must be distributed among the participants.

We consider the following three scenarios: decrease conference classification, increase conference classification, and change the classification to an incompatible level. Table 1 show our security analysis for these scenarios from the perspectives of security requirements for the call controller, active participants, new participants, and the need of new key generation.

	<b>Decrease conference level</b> $\lambda(\text{conf}_0) > \lambda(\text{conf}_{\text{new}})$	<b>Increase conference level</b> $\lambda(\text{conf}_{\text{new}}) > \lambda(\text{conf}_0)$	<b>Change to non-compatible level</b> $\lambda(\text{conf}_{\text{new}}) \not\geq \lambda(\text{conf}_0)$ $\lambda(\text{conf}_0) \not\geq \lambda(\text{conf}_{\text{new}})$
<b>Security requirement for call controller (<math>U_0</math>)</b>	<b>Trusted Subject</b>	$GLB[\lambda(T_0), \lambda(U_0, \text{password}_0)] \geq \lambda(\text{conf}_{\text{new}})$ must hold to authorize the change	<b>Trusted Subject and</b> $GLB[\lambda(T_0), \lambda(U_0, \text{password}_0)] \geq LUB[\lambda(\text{conf}_0), \lambda(\text{conf}_{\text{new}})]$ must hold to authorize the change
<b>Security requirement for active user (<math>U_i</math>)</b>	<b>None</b> , since $GLB[\lambda(T_i), \lambda(U_i, \text{password}_i)] \geq \lambda(\text{conf}_0) > \lambda(\text{conf}_{\text{new}})$	$GLB[\lambda(T_i), \lambda(U_i, \text{password}_i)] \geq \lambda(\text{conf}_{\text{new}})$ must hold not to be dropped	$GLB[\lambda(T_i), \lambda(U_i, \text{password}_i)] \geq LUB[\lambda(\text{conf}_0), \lambda(\text{conf}_{\text{new}})]$ must hold not to be dropped
<b>Security requirement for new user (<math>U_x</math>)</b>	$GLB[\lambda(T_x), \lambda(U_x, \text{password}_x)] \geq \lambda(\text{conf}_{\text{new}})$ must hold to join	$GLB[\lambda(T_x), \lambda(U_x, \text{password}_x)] \geq \lambda(\text{conf}_{\text{new}})$ must hold to join	$GLB[\lambda(T_x), \lambda(U_x, \text{password}_x)] \geq \lambda(\text{conf}_{\text{new}})$ must hold to join
<b>Need of new message encryption key distribution</b>	<b>YES</b> if a new participant with $\lambda(\text{conf}_{\text{new}}) \leq \lambda(U_x) < \lambda(\text{conf}_0)$ joins the conference <b>NO-if no new joins</b>	<b>YES</b> if any participants with $\lambda(U_i) < \lambda(\text{conf}_{\text{new}})$ has dropped out (voluntarily or forced) <b>NO-if no drops</b>	<b>YES</b> if a new join with $\lambda(\text{conf}_{\text{new}}) \leq \lambda(U_x)$ but $\lambda(\text{conf}_0) \leq \lambda(U_x)$ or if any participants with $\lambda(\text{conf}_{\text{new}}) \leq \lambda(U_i)$ has dropped out <b>NO-if no new joins and drops</b>

**Table 1.** Conference dynamics and security requirements

Note, that any change in the conference classification can be modeled as a series of single steps in the security lattice. That is, a change from label  $\lambda_1$  to  $\lambda_k$  is modelled as navigating the security lattice along the path  $\lambda_1 \rightarrow \lambda_2 \rightarrow \dots \rightarrow \lambda_k$ , where for all  $\lambda_i \rightarrow \lambda_j$  either  $\lambda_i > \lambda_j$  or  $\lambda_j > \lambda_i$ . For a call master to initiate the change of a conference level from  $\lambda_1$  to  $\lambda_k$  must be cleared to all intermediate levels, that is  $GLB[\lambda(T_0), [\lambda(U_0, \text{password}_0)]] \geq LUB[\lambda(\text{conf}_0), \lambda(\text{conf}_{\text{new}})]$  must hold. Similar restrictions hold for any active participant. Our analysis on the need of new encryption key incorporates the possibility that any non-active user may be eavesdropping on the conference before or after the change. The requirements for distributing a new key are based on this possibility of eavesdropping. Application requirements may require periodic refreshment of the message encryption key even if this is not necessary based on the conference dynamics.

## 4 Performance Analysis

We compute the delays of our protocol, using standard telecommunication connections delays [16,17,18], published encryption/decryption delays for text [3], and the switch response time delays. Table 2 in Appendix B summarizes our findings. The encryption and decryption time for RSA encryption and decryption is considered to be 12ms, (we do not consider the possibility of a small public key, therefore the encryption and decryption time is about the same). Table 3 in Appendix B shows the network delays corresponding to our protocols. The delays corresponding to the user interaction (like the time before an user answer the phone, the time necessary for a user to enter the password, or playing messages) are hard to measure and are user dependent, therefore are not considered here. The user interaction delay may take considerable time, but it is unavoidable and also part of traditional (un-secure) teleconferencing. The worst case calculation, given in Table 3, shows that teleconference setup delay is slightly less than 20 seconds under the assumption that all slave conferees are authenticated simultaneously (i.e., parallel authentication). Adding a user delay is about 11 seconds. Dropping a user and changing the conference classification create small (2-3 seconds) delays.

## 5 Conclusions

In this paper we present an architecture and protocols to facilitate multilevel secure teleconferences over Public Switched Telephone Network (PSTN). Our goal is to protect conversation confidentiality. Our protocols enable to establish secure telephone conferencing at a specific security level, add and drop conference participants, change the security level of an active conference, and tear down a conference. The protocols protect against eavesdropping and unauthorized participation in a MLS conference. MLS requirements are enforced by safeguarding the message encryption key of the conference. We also provide an initial estimates of delays incurred during setup (20 seconds) and adding a user (11 seconds).

The authors are not aware of any published acceptance delay range for automated teleconferencing. Based on our experiences using such services (e.g., observed delays of several minutes for conference set up) indicates that the delays, incurred by our protocols, are within the acceptable range. Nevertheless, for future references, we are planning to request evaluation of our protocols by vendors and developers. For future work we are planning to simulate our protocols to generate realistic measurements over the incurred delays. Furthermore, we are investigating methods to include a protocol for negotiating encryption algorithms, keys, and configurations specifications between the participants.

## 6 Acknowledgement

Farkas' work was partially supported by the National Science Foundation under Grant IIS-0237782.

## References

1. D. Bell and L. Lapadula. Secure computer systems : Unified exposition and multics interpretation. Technical Report ESD-TR-75-306, MITRE Corporation, 1975.
2. SecureLogix Corporation. TeleVPN call shield 1.0. <http://www.securelogix.com/applications/televpn.htm>.
3. Department of Defense Security Institute, <http://www.tscm.com/STUIIIhandbook.html>. *STU-III Handbook for Industry*.
4. ITU-T Recommendation Q.706. *Specifications of Signaling System No. 7-Message Transfer Part Signaling Performance*, March 1993.
5. ITU-T Recommendation Q.706. *Specifications of Signaling System No. 7-Signaling performance in the Telephone Application*, March 1993.
6. ITU-T Recommendation Q.709. *Specifications of Signaling System No.7-Hypothetical Signaling Reference Connection*, 1993.
7. ITU-T Recommendation Q.734. *Stage 3 description for multiparty supplementary Specifications of signaling system no. 7.*, 1993.
8. ITU-T Recommendation Q.84. *Stage 2 description for multiparty supplementary services*, 1993.
9. ITU-T Recommendation Q.954. *Stage 3 description for multiparty supplementary services using DSS 1*, 1993.
10. G. Lorenz, T. Moore, J. Hale, and S. Sheno. Securing SS7 telecommunications networks. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, 2001.
11. T. Russell. *Signaling system 7*. McGraw-Hill, New York, 2002.
12. M. Sharif and D. Wijesekera. Providing voice privacy over public switched telephone networks. In *Proceeding of IFIP 11.5*, pages 25–36, 2003.
13. J. G. von Bosse. *Signaling in Telecommunication Networks*. John Wiley & Sons, New York, 1998.
14. I. Youn and D. Wijesekera. Secure bridges: A means to conduct secure teleconferences over public telephones. In *Proc. of the 18th Annual Conference on Data and Applications Security*, 2004.

## Appendix A

### Protocol 1 – Conference Set Up:

#### A. Call controller authentication:

1. [ $T_0$ ] The call controller ( $U_0$ ) dials the teleconference access code. Once the telephone enters the teleconference mode, the call controller enters the telephone line number ( $TLN$ ) of the master secure bridge ( $MSB$ ).
2. [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ]  $T_0$  invokes the facility that initiates the conference sending  $M_1 = K_{msb}[K_0^*[TLN_0, TDN_0, t_0]]$  to  $MSB$ , where  $K_{msb}$  is the public key of the  $MSB$ , and  $K_0^*$  is the private key of  $T_0$ . This message is used for the authentication of the telephone device and travels in the SETUP message (ISDN) between  $T_0$  and  $SSP_0$ , and in the IAM primitive (ISDN) between  $SSP_0$  and  $SSP_{msb}$ . While the IAM message travels through the SS7 network, the intermediate exchanges allocate the voice trunks. The destination exchange ( $SSP_{msb}$ ) allocates the resources for the secure teleconference (the Master Secure Bridge - MSB) and initiates the teleconference transaction by sending the message  $M_1$  to  $AC_{msb}$ .

3. [ $AC_{msb}$ ] The authentication center of the Master Secure Bridge verifies the authenticity of the telephone set by extracting  $TLN_0$  and  $TDN_0$  and comparing them against the ones stored in the database, and by checking also the validity of the timestamp to prevent the replay attack. The authentication center looks in its database for the telephone classification  $\lambda(T_i)$  and associates it with the initiated teleconference.

**If** authentication succeeds and the algorithm continues with the step A.4.

**Else**,  $SSP_m$  clears the allocated voice trunks using a RELEASE/RLCOM message pair that propagates along the allocated path.

4. [ $AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0$ ]  $MSB$  replies with a request for user authentication embedded in a Call Progress (CPG) message:

$M_2 = K_b^* r[ID\_Request, R_0^*, t_1]$ , where  $R_0^*$  is a nonce generated by  $AC$  that will be embedded in the message exchanged between call master and  $MSB$  during the teleconference session, and  $t_1$  is a timestamp. Both the random number and the timestamp are meant to prevent the replay attack. An IVR message solicits the user to dial her user ID.

5. [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ] The call controller enters her ID ( $ID_0$ ):  
 $M_3 = K_{msb} [ID_0, R_0^*, t_3]$

6. [ $AC_{msb}$ ] The authentication center of the  $MSB$  decrypts  $M_3$  and checks the validity of the random number, timestamp, and looks in the database for  $ID_0$ .

**If** authentication succeeds the protocol continues with the step A.7.

**Else**,  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair and ends the transaction with  $AC_{msb}$

7. [ $AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0$ ] The authentication center sends a signed acknowledgement in a CPG message, which contain a request for password:  $M_4 = K_{msb}^* [PWD\_Request, R_0^*, t_4]$ .

8. [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ] The call controller dials her password ( $PWD_0$ ), which will be again send to  $AC_{msb}$  in a CPG message encrypted by the public key of  $MSB$ .  $M_5 = K_{msb} [PWD_0, R_0^*, t_5]$

9. [ $AC_{msb}$ ] The  $MSB$  decrypts the message and checks the timestamp and the ( $ID_0, PWD_0$ ) pair.

**If** authentication succeeds, i.e., there is an ( $ID_0, PWD_0$ ) pair, the  $AC_{msb}$  maps the user clearance  $\lambda(U_0)$ .  $AC_{msb}$  computes  $\lambda(permitted) = GLB[\lambda(T_i), \lambda(U_0)]$ . The protocol continues with step 10.

**Else**,  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair and ends the transaction with  $AC_{msb}$ .

## B. Conference classification and the telephone line numbers

1. The call master dials the number of the  $nc$  conferees, one by one ( $nc$  is a number between 1 and 30). We suppose that only  $n$  conferees ( $n = nc$ ) succeed in connecting to the conference. The other ( $nc - n$ ) conferees do not connect or have authentication failure.

2. [ $AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0$ ] The authentication center requests the call master to choose a classification for the conference (LC):  $M_6 = K_{msb}^* [LC\_Request, R_0^*, t_6]$ . This is requested as a list of options played using the IVR.

3. [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ] The call controller sends the classification for the conference:  $M_7 = K_{msb} [\lambda(conference), R_0^*, t_7]$ .

**If**  $\lambda(conference) \leq \lambda(permitted)$  then the protocol continues with step B.4.

**Else**  $SSP_{msb}$  clears the allocated voice trunks using a RELEASE/RLCOM message pair and ends the transaction with  $AC_{msb}$ .

4. The following steps are repeated for all remote parties ( $i = 1, 2 \dots n$ )
  - (a) [ $AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_0 \rightarrow T_0$ ] *MSB* sends a request to the call controller to dial the telephone line number of the first conferee:  
 $M_8 = K_{msb}[TLN\_request, R_0^*, t_{8i}]$
  - (b) [ $T_0 \rightarrow SSP_0 \rightarrow SSP_{msb} \rightarrow AC_{msb}$ ] The call controller dials the telephone line number of *User<sub>i</sub>* ( $TLN_i$ ):  $M_9 = K_{msb}[TLN_i, R_0^*, t_{9i}]$
5. For  $i = 1$  to  $nc$  repeat the following steps (1 through 9) ( $nc$  is the number of conferees called by  $U_0$ ). If  $\lambda(conference) \leq \lambda(permitted)$  for user  $U_i$  than associate ( $U_i, \lambda(permitted)$ ) with the conference and continue the protocol. Else drop  $U_i$  and clear the connection.

### C. Cross-certification

1. [ $AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_i$ ]  $AC_{msb}$  signals to  $SSP_{msb}$  to send the initial address message (IAM) that seizes a trunk between the secure bridge for  $U_i$  and the local exchange of the remote user ( $SSP_i$ ) to establish a bidirectional circuit between the secure bridge and the  $SSP_i$ , followed by a call progress (CPG) message that has as a parameter a ticket  $M_{10} = K_{msb}^*[ID_0, ID_1 \dots ID_{nc}, \lambda(conference), R_i, t_{10i}]$  signed by the bridge.  $M_{10}$  certifies the  $U_0$ , initiate the conference, and transmits the conference classification  $\lambda(conference)$  to the *SSBs*.
2. [ $SSP_i \rightarrow AC_i$ ]  $SSP_i$  forwards  $M_{10}$  to  $AC_i$  for authentication. If fails, the  $AC_i$  signals the  $SSP_i$  to drop the *User<sub>i</sub>*. Otherwise, continues with step D.1.

### D. Remote parties authentication

1. [ $AC_i \rightarrow SSP_i \rightarrow T_i \rightarrow SSP_i$ ] If authentication succeeds,  $AC_i$  sends authentication result to the  $SSP_i$  in TCAP message  $M_{11} = K_{aci}^*[ID_0, ID_1 \dots ID_{nc}, K_{msb}, R_i^*, t_{11i}]$ .  $SSP_i$  sends the result to the  $T_i$  in an ISUP message.  $T_i$  sends back:  $M_{12} = K_{aci}[K_i^*[TLN_i, TDN_i, R_i^*, t_{12i}]]$  encrypts and signs telephone device and line numbers.
2. [ $SSP_i \rightarrow AC_i$ ] The authentication center checks the telephone line and the device numbers in  $M_{12}$  sent through the TCAP message by decrypting the message with  $K_{aci}^*$ , and then checks the signature of  $T_i$  using  $K_i$ . After decryption and authentication, the  $AC_i$  also verifies whether the  $TLN_i$  and the  $TDN_i$  from the message  $M_{12}$  coincides with the one in the local database. Also,  $AC_i$  looks in its database for the telephone device classification  $\lambda(T_i)$ . If the authentication fails, or if the security condition  $\lambda(conference) \leq \lambda(T_i)$  fails,  $AC_i$  sends an error message to the  $SSP_i$ , which initiates the disconnection procedure for the *User<sub>i</sub>* from the secure conference by sending a REL/RLCOM message pair to the *MSB*.
3. [ $AC_i \rightarrow SSP_i \rightarrow T_i$ ]  $AC_i$  sends  $M_{13} = K_{aci}^*[ID\_Request, R_i^*, t_{13i}]$  in a TCAP message as the return result to the  $SSP_i$  where the random number  $R_i^*$  is included in the confirmation ticket sent by the  $AC_i$  to the *MSB*.
4. [ $SSP_i \rightarrow T_i$ ]  $SSP_i$  sends  $M_{13}$  to  $U_i$  in a FACILITY message with a FIE containing a user authentication request.
5. [ $T_i \rightarrow SSP_i$ ]  $T_i$  sends the ALERT ( $CR_i$ ) message to  $SSP_i$ .
6. [ $SSP_i \rightarrow SSP_0$ ]  $SSP_i$  sends the ALERT
7. [ $SSP_0 \rightarrow T_0$ ]  $SSP_0$  sends the ALERT ( $CR_0$ ) message to  $T_0$ .
8. [ $T_i \rightarrow SSP_i$ ] When  $U_i$  picks up the handset,  $T_i$  sends the CONNECT message to  $SSP_i$ , and the  $SSP_i$  plays an IVR announcement informing  $U_i$  of the conference participants, after the  $SSP_i$  plays a new IVR announcement to the  $T_i$ : "Please enter your ID".

9.  $[T_i \rightarrow SSP_i]$   $U_i$  dials her ID that is encrypted with  $AC_i$ 's public key. The  $U_i$ 's telephone knows the  $AC_i$ 's public key, and sends it to the  $AC_i$  over the network.
10.  $[SSP_i \rightarrow AC_i]$   $SSP_i$  forwards  $M_{14} = K_{aci}[ID_i, R * i, t_{14}]$  to the  $AC_i$  in a TCAP message. The authentication center verifies the pair  $(ID, password)$  sent over by the  $SSP_i$ . If the ID is not found in  $AC_i$ 's database, or if the condition  $\lambda(conference) = \lambda(T_i)$  is not fulfilled, the  $AC_i$  issues an error message to the  $SSP_i$ , and the local exchange starts clearing the connection. Thus we have:  $\lambda(conference) = GLB[\lambda(T_i), \lambda(U_i)] = \lambda(T_i, U_i)$
11.  $[AC_i \rightarrow SSP_i \rightarrow T_i]$  If the authentication succeeds,  $AC_i$  sends a *PWD* request to the  $User_i$ :  $M_{15} = K^*_{aci}[PWD\_Request, R_i^*, t_{15i}]$
12.  $[T_i \rightarrow SSP_i \rightarrow AC_i]$   $User_i$  answers with  $M_{16} = K^*_{aci}[PWD_i, R_i^*, t_{16i}]$
13.  $[AC_i]$   $AC_i$  checks the password, and if authentication fails, clears the connections with *MSB* and  $T_i$ . If authentication succeeds, it continues with step E.1.

### E. Cross-certification

1.  $[AC_i \rightarrow SSP_i \rightarrow SSP_{msb} \rightarrow AC_{msb}]$  If authentication succeeds,  $SSP_i$  sends the following ticket to  $AC_{msb}$ , completing the cross-certification phase:  $M_{17} = K^*_{aci}[ID_i, \lambda(U_{17i})]$ . The *MSB* receives now  $U_i$ 's public key and clearance, and also the telephone device classification. Thus, *MSB* and  $U_i$  are able to communicate without any further help from the slave secure bridge. *MSB* double-checks the condition  $\lambda(conference) = GLB(\lambda(T_i), \lambda(U_i)) = \lambda(T_i, U_i)$

### F. Key Distribution

1. The master secure bridge waits until either all users have connected or a connection timeout occurred, and adds the IDs of all connected users to a list.
2. For  $i = 0, 1, \dots, n$  repeat following steps 1 and 2.
  - (a)  $[AC_{msb} \rightarrow SSP_{msb} \rightarrow SSP_i \rightarrow T_i]$ . The secure bridge starts the group shared key distribution phase by sending  $M_{18} = K_i[K_{br}^*[K_E, R_i^*, t_{18}]]$  in a TCAP message between the  $AC_{msb}$  and the  $SSP_{msb}$ , in a CPG message between the  $SSP_{msb}$  and the  $SSP_i$  and in a FACILITY message between  $SSP_i$  and  $T_i$ .
  - (b)  $[T_i \rightarrow SSP_i \rightarrow SSP_{msb} \rightarrow AC_{msb}]$   $T_i$  decrypts  $M_{18}$ , checks the signature, the random number and the timestamp, and recovers the group shared key  $K_E$ . After this, the  $T_i$  sends the *Key - dist - ACK*( $ID_i$ ) back to the  $AC_{msb}$ .
3. As soon as the users receive the symmetric key, they can start the secure group conversation. The voice is encrypted by the telephone device and is sent to the Master Secure Bridge. The *MSB* takes care of forwarding the encrypted signal to the destination telephone devices, where the signal is decrypted.

## Appendix B

Type of Call Segment	Switch Response time (ms)	
	Mean	95% confidence interval
ISUP Message	205 – 218	= 337 – 349
Alerting	400	= 532
ISDN Access Message	220 – 227	= 352 – 359
TCAP Message	210 – 222	= 342 – 354
Announcement/Tone	300	= 432
Connection	300	= 432
End MF Address - Seize	150	= 282

**Table 2.** Switch Response Delay Calculation

Table Conference Call Phase	Delay	Delay under assumptions: $n = 10, p = 10s,$ $a_i = b_i = 50ms$ $d = e = 12ms$	Description of the parameters and assumptions
Call setup	$11007 + 667n$ $+(n + 8)a_0$ $+8 \cdot \max\{a_1 \dots a_n\}$ $+2 \cdot \max\{b_1 \dots b_n\}$ $+21 \cdot (d + e)$ ms	19, 181ms	The number of conferencing subscribers is $n$ The transmission propagation delay between $T_0$ and $AC_{msb}$ is $a_0$ and the transmission propagation
Add user by call controller	$9855 + 3a_0 + 6a_n$ $+2 \cdot \max a_1 \dots a_n$ $+2 \cdot \max b_1 \dots b_n$ $+11 \cdot (d + e)$ ms	10, 769ms	delay between $T_i$ and $AC_i$ is $a_i$ , where $i = 1, 2 \dots n$ . (see ITU-T Recommendation TABLE 1/Q.706). We will omit a maxi-
Drop user by call controller	$2001 + a_0$ $+2 \cdot \max\{a_1 \dots a_n\}$ $+2 \cdot \max\{b_1 \dots b_n\}$ $+3 \cdot (d + e)$ ms	2, 323ms	imum 2.5ms delay between $T_0$ and $SSP_0$ (under the realistic assumption that the distance between $T_0$ and $SSP_0$ is less then 500km),
Increase / change conference classification.	$2001 + a_0$ $+2 \cdot \max a_1 \dots a_n$ $+2 \cdot \max b_1 \dots b_n$ $+3 \cdot (d + e)$ ms	2,323 ms	since it is not significant compared with the total delay. The transmission propagation delay between $AC_{msb}$ and $AC_i$ is $b_i$ , where $i = 1, 2 \dots n$
Decrease conference classification	$2001 + a_0$ $+2 \cdot \max\{a_1 \dots a_n\}$ $+2 \cdot \max\{b_1 \dots b_n\}$ $+(d+e)$ ms	2, 275ms	The delay to perform a RSA 1024 encryption/decryption is $e = d = 12ms$ .

**Table 3.** Network delay