International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:10, 2007

# Dynamic Authenticated Secure Group Communication

R. Aparna, and B. B. Amberker

***Abstract*—**Providing authentication for the messages exchanged between group members in addition to confidentiality is an important issue in Secure Group communication. We develop a protocol for Secure Authentic Communication where we address *authentication* for the group communication scheme proposed by Blundo et al. which only provides *confidentiality*. Authentication scheme used is a multiparty authentication scheme which allows all the users in the system to send and receive messages simultaneously. Our scheme is secure against colluding malicious parties numbering fewer than *k*.

***Keywords*—**Secure Group Communication, Secret key, Authentication, Authentication code, Threshold.

## I. Introduction

THE availability of digital technologies and widening internet bandwidth has increased the demand for new multimedia services. Some of the service types include video-on-demand, scientific discussion, board meeting, real-time delivery of stock market information. There are many users who take part in the services like teleconferencing, board meeting, scientific discussion etc. The communication among the users pertaining to one service must be carried out confidentially.

Other scenario may be out of $n$ users in a network, some $t$ ($t<<n$) of them would like to discuss on a common concern. These $t$ parties termed *privileged users* must communicate themselves over a public channel and others must not be able to listen to the conversation between these $t$ parties. Hence, there is a need to find new technology for such confidential communication termed as *Secure Group Communication* or *Secure Conferencing*.

A Naïve solution is to have a shared key between every pair of users, which leads to storing $(n-1)$ keys with each user. Also, to send a message, sender must encrypt the message to each user in the group separately. This increases the amount of storage at each user and also computation and communication costs are increased. Hence, the general goal of Secure Group Communication is to establish a *common secret key*, also called *Secure Group key* or *Secure Conference key* among privileged users for confidential communication.

R. Aparna is with Department of Computer Science and Engineering, Siddaganga Institute of Techology, Tumkur, Karnataka, India (e-mail: raparna@sit.ac.in).

B. B. Amberker is with Department of Computer Science and Engineering, National Institute of Technology, Andhra Pradesh, Warangal, India (e-mail: bba@nitw.ac.in).

Once a conference (group key) is set up, users in the group can communicate with each other securely. Since the group is dynamic, members in the group may change over time, i.e., new members may join the group and existing members may leave the group. Group membership can change because a single member joins/leaves the group or a set of members may join/leave the group simultaneously. Whenever there is a membership change, group key must be changed to prevent a new user from reading past communications, called *backward access control* and a departed user from reading future communications, called *forward access control*.

In Key management schemes, an adversary may try to eavesdrop on the conversation of the privileged user set. An adversary may be an insider (member among $n$ parties, but not a member of privileged set) or outsider (member other than $n$ parties). A group of users, termed malicious parties may collude with each other and try to derive the common group key. The security of the group key management scheme is based on number of colluding parties. Scheme is termed as *k*-secure, if it is not possible to derive the common key even after $k$ non-group (non-privileged) members collude with each other, where $k$ is termed as *threshold*.

This type of communication among group members using common secret key in a secure group communication model will ensure only confidentiality of the message. In any secure group communication scenario, since all the group members can send and receive messages, the sender of a message must be able to indicate his identity and the receivers must be able to verify the authenticity of the message. An authentication service is concerned with assuring that the communicating entity is the one that it claims to be. Authentication code is generated and sent along with the message by the sender and the same is used by the receivers to verify authenticity of the received message.

In a conventional point-to-point authentication system [16], each pair of users is given with a shared key. To broadcast an authenticated message, user will construct a separate authenticator for every other user, concatenates them and will append it to the message. This method increases the amount of key storage at each user, produces a very long authentication tag for the message which results in high communication cost.

Desmedt, Frankel and Yung developed an authentication scheme in [5] which considers a single transmitter who is fixed before hand. Initially a trusted authority distributes secret key information to all $n$ users in the system. A transmitter broadcasts a message to $n$ receivers in the system who will individually verify the authenticity of the message using their secret key information which was previously

distributed by the trusted authority.

Safavi-Naini and Wang extended the scheme of Desmedt, Frankel and Yung in [13], [14] and [15]. In [13] and [14] they relaxed the restriction that the sender is fixed before hand and introduced a dynamic sender concept in which any one of the users can become the sender after the initial stage of key distribution by the trusted authority.

In [15], they dropped the restriction of a single sender and proposed a scheme for the situation with t senders. This scheme uses symmetric polynomials in two variables over GF(q) and is developed from Blom's key distribution scheme [2]. Here the size of the secret key information at each user, and of the authentication tag for a message, grew linearly with $t$.

In [1] we have developed a multiparty authentication scheme in which we have considered the setup in [14, 15], but dropped the restriction on the number of senders. That is, in our scheme we allow some (or all) of n parties $User_1, \cdots, User_n$ to send and receive messages. The scheme provides authenticity for the messages exchanged between a group of $n$ users. In this scheme each user is required to store secret information of size $2klog_2q$ bits and tags to authenticate messages are of length $klog_2q$ (Throughout the paper q is a prime power such that q ≥ size of message space. We assume that $k$ and $n$ are such that size of message space ≥ $2kn$). Here, the size of the authentication tag are not dependent on the number of senders. In addition to this, each user is required to store a further $8(n-1)klog_2q$ bits of information which is public. The security of the scheme is indifferent to exposure of this further information, either to an adversary or to the other participants.

A group of malicious parties-who number fewer than $k$ (where $k$ is the threshold) -may collude and try to launch an attack (by using their secret keys and all previous communications) against a pair, say $User_i$ and $User_j$ by generating a message such that $User_j$ accepts it as authentic and being sent from $User_i$. We have derived schemes in which perfect protection is guaranteed against such attacks.

Since the scheme provides perfect protection against colluding malicious parties numbering fewer than $k$ and allows all the $n$ users in the system to be senders (simultaneously being receivers), the scheme is more appropriate for providing authentication in secure group communication and also since the secret storage at each user is independent of the value of $n$, it is applicable even for a secure group with large number of users.

Several Group Key management techniques have been proposed in [3, 11, 6, 8, 9, 7, 4, 10, 12]. All these schemes address computation of secure group key for confidential communication among the group members. Among these schemes some are *information theoretic* and some being *complexity theoretic*. Most of the schemes involve a single entity termed as *Key Distribution Center* (KDC), which is responsible for generating and distributing private initial pieces of information to all the users in the system. After receiving initial pieces of information from KDC, group members may compute the group key either *non-interactively* or *interactively*. In non-interactive, each member in the secure group can derive a common group key on his own with the help of information obtained by KDC. On the other hand, in interactive, users in the secure group communicate with each other to set up the group key.

In this paper we consider an information theoretic based group key management scheme proposed by Blundo et al. [3]. In [3], a non-interactive protocol has been developed to derive a common group key for secure communication. It provides only confidentiality of the messages exchanged between the users of the secure group. In this paper we propose an authentication scheme which can be applied to provide authentication for the protocol proposed by Blundo et al. in [3]. The rest of the paper is organized as follows: Section 2 describes the model used, Section 3 gives an insight into Blundo et al. conference keying protocol. In Section 4, we discuss message authentication scheme and Section 5 focuses on providing authentication to Blundo et al. secure group communication model. Section 6 concludes the paper.

## II. Model

Our secure group communication model comprises of a total of $n$ users $User_1, User_2, \cdots, User_n$ out of which $t$ ($t \leq n$) users can form a secure group. Value of $t$ is fixed before hand, but out of n users, any $t$ of them can participate in the conference and these $t$ users may vary dynamically. These $t$ users in the secure group can compute a secure group key non-interactively by using Blundo et al. non-interactive $k$ secure $t$-conference protocol as in [3].

Our authentication scheme contains three phases:

**1. Key Distribution:** The trusted KDC picks and distributes private key information to all the users in secret.

**2. Broadcast:** The sender broadcasts a message to all the other users in the system, along with an authentication tag.

**3. Verification:** Each user verifies the authenticity of the message broadcast by the sender.

This scheme provides a perfect protection against collusion of up to k members in which senders and receivers use an authentication tag to verify the authenticity of the message received.

Secure Authentic Communication scheme discussed in this paper comprises of seven phases: Polynomial Selection, Key Distribution, Polynomial Construction for Authentication, Computation of $\alpha_{i,j}$ and $\beta_{i,j}$, Conference Key Computation, Secure Authentic Communication and Verification.

## III. Blundo et al. Conference Keying Scheme

Blundo et al. [3] have proposed a protocol to derive a common conference key. In this approach, a trusted off line server which is active only at initial stage of the protocol distributes some information among a set of $n$ users, $User_1, User_2, \ldots, User_n$ so that any $t$ of them can join and generate a secure group key non-interactively. Fig. 1 depicts the Blundo et al. non-interactive $k$-secure $t$-conference protocol. It makes use of a symmetric polynomial in $t$ variables (number of users in the group) of degree $k$ (threshold) with coefficients over GF($q$), $q > n$. In this scheme, $t$ is fixed before hand. Each

join/leave operation restarts the protocol from initial stage.

Maximum number of coefficients in the polynomial are $\binom{k+t-1}{t-1}$ out of which each user is required to store $\binom{k+t-2}{t-2}$ values from GF($q$) and is required to perform at the most $t*\binom{k+t-2}{t-2}$ multiplications, $\binom{k+t-2}{t-2}$ additions and $k*t$ exponentiation operations.

- KDC picks at random a symmetric polynomial P ($x_1$, . . . , $x_t$) of degree $k$ with $t$ variables with coefficients over GF ($q$), $q > n$.
- To each user $User_i$, i = 1, . . . , n, in the system, KDC distributes the polynomial $f_i(x_2, . . . , x_t) = P(i, x_2, . . . , x_t)$, that is the polynomial obtained by evaluating $P(x_1, . . . , x_t)$ at $x_1 = i$.
- If the users $User_{j1}$ , . . . , $User_{jt}$ want to set up a conference key then each user $User_{ji}$ evaluates $f_{ji}(x_2, . . . , x_t)$ at $(x_2, . . . , x_t) = (j_1, . . . , j_{i-1}, j_{i+1}, . . . , j_t)$.
- The conference key for users $User_{j1}$, . . . , $User_{jt}$ is equal to BK = $P(j_1, . . . , j_t)$.

Fig. 1 Blundo et al. non-interactive k-secure t-conference protocol

## IV. Message Authentication Scheme

Multiparty authentication scheme discussed in [1] provides authentication for the messages exchanged between a group of $n$ users, $User_1$, . . . , $User_n$. The scheme comprises of three phases: Key Distribution, Broadcast Encryption and Verification.

**Key distribution:** The (trusted) key distribution center picks at random $2nk$ elements $a_{11}, a_{12}, . . . , a_{1k}$ , $a_{21}, a_{22}, . . . , a_{2k}$ , . . ., $a_{n1}, a_{n2}, . . . , a_{nk}$, and $b_{11}, b_{12}, . . . , b_{1k}$, $b_{21}, b_{22}, . . . , b_{2k}$, . . ., $b_{n1}, b_{n2}, . . . , b_{nk}$, from GF ($q$). It sends the $2k$ elements $a_{i1}, a_{i2}, . . ., a_{ik}$, and $b_{i1}, b_{i2}, . . . , b_{ik}$ to $User_i$ in secret. Denote by $A_i(x)$ the polynomial $a_{i1}x^k + a_{i2}x^{k-1} + \cdots + a_{ik}x$, and by $B_i(x)$ the polynomial $b_{i1}x^k + b_{i2}x^{k-1} + \cdots + b_{ik}x$. For each ordered pair (i, j) ($i \neq j$) it determines an $\alpha_{i,j}$ satisfying the equation

$$\alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) = 1;$$

in other words, such that

$$a_{i1}\alpha_{i,j}^{2k} + a_{i2}\alpha_{i,j}^{2k-1} + \cdots + a_{ik}\alpha_{i,j}^{k+1} + a_{j1}\alpha_{i,j}^{k} + a_{j2}\alpha_{i,j}^{k-1} + \cdots + a_{jk}\alpha_{i,j} = 1.$$

It then determines the value $\beta_{i,j}$ taken by the polynomial $x^kB_i(x) + B_j(x)$ at $x = \alpha_{i,j}$, i.e.,

$$\beta_{i,j} = \alpha_{i,j}^k B_i(\alpha_{i,j}) + B_j(\alpha_{i,j})$$

(Notice that, since $x^kA_i(x) + A_j(x)$ is a polynomial of degree $2k$ over GF ($q$), $\alpha_{i,j}$ (therefore also $\beta_{i,j}$) may be taken to be elements of GF ($q^{2k}$).) It publicly sends to $User_i$ the values $\alpha_{i,j}$, $\alpha_{j,i}$, $\beta_{i,j}$ and $\beta_{j,i}$, j ∈ {1, . . . , n} \ {i}. (Thus any participant could find out all the $\alpha_{i,j}$'s and, $\beta_{i,j}$'s.)

**Broadcast:** $User_i$, in order to send a message s, broadcasts $(a_{i1} + sb_{i1}, a_{i2} + sb_{i2}, . . . , a_{ik} + sb_{ik}, s)$.

**Verification**: In effect $User_i$ has broadcast (s and) the polynomial $F_{s,i}(x) = A_i(x) + sB_i(x)$. On receiving this, $User_j$ forms the polynomial $x^k\{F_{s,i}(x)\} + A_j(x) + sB_j(x)$ using its secret information (which, in effect, consists of the polynomials $A_j(x)$ and $B_j(x)$). It verifies that this polynomial evaluates to $1 + s\beta_{i,j}$ at $x = \alpha_{i,j}$, which will indeed be the case, since

$$\begin{aligned}&\alpha_{i,j}^k\{F_{s,i}(\alpha_{i,j})\} + A_j(\alpha_{i,j}) + sB_j(\alpha_{i,j})\\ &= \alpha_{i,j}^k\{A_i(\alpha_{i,j}) + sB_i(\alpha_{i,j})\} + A_j(\alpha_{i,j}) + sB_j(\alpha_{i,j})\\ &= \alpha_{i,j}^kA_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) + s\{\alpha_{i,j}^kB_i(\alpha_{i,j}) + B_j(\alpha_{i,j})\}\\ &= 1 + s\beta_{i,j}\end{aligned}$$

**Storage required:** $User_i$ is required to hold the $2k$ values $a_{i1}, a_{i2}, . . . , a_{ik}, b_{i1}, . . . , b_{ik}$ ∈ GF ($q$) in secret. This comes to $2k \log_2 q$ bits of secret information. $P_i$ has to also store the $4(n - 1)$ elements $(\alpha_{i,j}, \alpha_{j,i}, \beta_{i,j}, \beta_{j,i})$ ∈ GF ($q^{2k}$), j ∈ {1, . . . , n} \ {i}, but these do not need to be guarded against exposure. This is a further $8(n - 1)k \log_2 q$ bits

**Proof of security:** Suppose $User_{i1}$ , $User_{i2}$ , . . . , $User_{i(k-1)}$, $i_1, . . . , i_{k-1}$ ∈ {1, . . . , n} \ {i, j}, is a group of malicious receivers who wish to impersonate $User_i$ while sending the message s' to $User_j$. In order to do so, they need to determine the $k$ coefficients of the polynomial $F_{s',i}(x) = A_i(x) + s'B_i(x)$.

The secret information they possess between them corresponds to the conditions

$$F_{s',i}(\alpha_{i,ir}) = \alpha_{i,ir}^{-k}\{1 + s'\beta_{i,ir} - A_{ir}(\alpha_{i,ir}) - s'B_{ir}(\alpha_{i,ir})\}$$

for r = 1, . . . , $k - 1$. Since these fix the evaluations of $x^{-1}F_{s',i}(x)$, which is known only to be a polynomial of degree $= k - 1$ and on whose coefficients there are no further restrictions, only at $k - 1$ points, they only determine a set of $q$ polynomials (of degree $= k - 1$) to which $x^{-1}F_{s',i}(x)$ belongs.

## V. Secure Authentic Communication

In order for the users in the secure group to communicate with confidentiality and authenticity, we can combine the features of authentication scheme explained in section 4 with that of Blundo et al. [3] conference keying protocol discussed in section 3. Protocol in Fig. 2 demonstrates authenticated secure group communication for a group with $t$ users.

In the protocol each user $User_i$, i = 1, . . . , $n$ is required to store in secret $\binom{k+t-2}{t-2}$ values and 2k elements from GF ($q$) and also 4(n-1) elements grom GF($q^{2k}$), of public information.

If ($k - 1$) malicious parties $User_{i1}$, $User_{i2}$, . . . , $User_{i(k-1)}$ , $i_1, . . . , i_{k-1}$ ∈ {1, . . . , n} \{i, j} collude and try to impersonate $User_i$ while sending a message to $User_j$, $User_j$ fails to verify

the authenticity of $User_i$ as per the proof of security illustrated in Section 4. Hence the scheme is secure against fewer than $k$ malicious parties and provides confidential authentic communication between group members which is more appropriate for the applications like scientific discussion, board meeting etc.

## VI. Conclusion

Providing confidentiality for the messages exchanged between users of a particular group is an important issue. Providing authenticity for this confidential communication between group members is as important as providing confidentiality. The authentication scheme proposed in this paper is secure against collusion of fewer than $k$ malicious parties and fits very well for secure group communication scenario even with large number of users, since the storage required at each user is independent of group size.

- **Polynomial Selection:** KDC picks at random a symmetric polynomial $P(x_1,\ldots,x_t)$ of degree $k$ with $t$ variables and $2nk$ elements $a_{11},a_{12},\ldots,a_{1k},\ a_{21},a_{22},\ldots,a_{2k},\ \ldots,\ a_{n1},a_{n2},\ldots,a_{nk}$, and $b_{11},b_{12},\ldots,\ b_{1k},\ b_{21},b_{22},\ \ldots,b_{2k},\ \ldots,\ b_{n1},b_{n2},\ldots,b_{nk}$. Coefficients of the polynomial and $2nk$ elements are from $GF(q)$, $q>n$.
- **Key Distribution:** To each user $User_i$, $i=1,\ldots,n$, in the system, KDC distributes $2k$ elements $a_{i1},a_{i2},\ldots,\ a_{ik}$, and $b_{i1},b_{i2},\ldots,b_{ik}$ i.e., the polynomials $A_i(x)=a_{i1}x^k+a_{i2}x^{k-1}+\cdots+a_{ik}x$, and $B_i(x)=b_{i1}x^k+b_{i2}x^{k-1}+\cdots+b_{ik}x$ and the polynomial $f_i(x_2,\ldots,x_t)=P(i,x_2,\ldots,x_t)$, that is the polynomial obtained by evaluating $P(x_1,\ldots,x_t)$ at $x_1=i$.

  For each ordered pair $(i,j)$ $(i\neq j)$ it determines an $\alpha_{i,j}$ satisfying the equation
  $$\alpha_{i,j}^k A_i(\alpha_{i,j})+A_j(\alpha_{i,j})=1;$$
  and determines the value $\beta_{i,j}$ taken by the polynomial $x^kB_i(x)+B_j(x)$ at $x=\alpha_{i,j}$, *i.e.*,
  $$\beta_{i,j}=\alpha_{i,j}^k B_i(\alpha_{i,j})+B_j(\alpha_{i,j})$$
- **Conference Key Computation:** If the users $User_{j_1},\ldots,User_{j_t}$ want to set up a conference key then each user $User_{j_i}$ evaluates $f_{j_i}(x_2,\ldots,x_t)$ at $(x_2,\ldots,x_t)=(j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_t)$.

  The conference key for users $User_{j_1},\ldots,User_{j_t}$ is equal to $BK=P(j_1,\ldots,j_t)$.
- **Secure Authentic Communication:** If $User_i$ wants to send a message m securely, it broadcasts $(a_{i1}+mb_{i1},a_{i2}+mb_{i2},\ldots,a_{ik}+mb_{ik},E_{BK}(m))$ i.e., it broadcasts the polynomial $F_{m,i}(x)=A_i(x)+mB_i(x)$ along with encrypted message.
- **Verification:** Upon receiving this information, $User_j$ first decrypts $E_{BK}(m)$ using conference key $BK$ to get the message $m$. Now, $User_j$ constructs the polynomial $x^k\{F_{m,i}(x)\}+A_j(x)+mB_j(x)$ and verifies that this polynomial evaluates to $1+m\beta_{i,j}$ at $x=\alpha_{i,j}$.

  Hence $User_j$ has verified the authenticity of the sender i.e., $User_i$.

Fig. 2 Protocol for Secure Authentic Group Communication

## References

[1] R. Aparna, B.B. Amberker, Prashant Koulgi. Multi-sender Multi-receiver Message Authentication and an Application to Verifiable Secret Sharing. National conference on Mathematical Foundations of Coding, Complexity, Computation and Cryptography, IISc., Bangalore , June 2006.

[2] Blom, R.: An Optimal Class of Symmetric Key Generation Systems, in Advances in Cryptology-Eurocrypt '84, LNCS, 209 (1985), 335-338.

[3] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung. Perfectly Secure Key Distribution for Dynamic Conferences, in Advances in Cryptology-CRYPTO'92.

[4] M. Burmester. and Y. Desmedt, A Secure and Efficient Conference Key Distribution System, Advances in Cryptology - EUROCRYPT'94.

[5] Desmedt, Y., Frankel, Y., Yung, M.: Multi-receiver/Multi-sender Network Security: Efficient Authenticated Multicast/Feedback, IEEE Infocom '92, pp.2045-2054, (1992).

[6] W. Diffie and M. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, IT-22(6):644-654, Nov 1976.

[7] A. Fiat and M. Naor, Broadcast Encryption. In D.R.Stinson, editor, Proceedings of CRYPTO'93, pp 480-491.

[8] A.T. Sherman and D.A. McGrew. Key Establishment in Large Dynamic groups using One-way Function trees. IEEE Transactions on Software Engg. 2003, pp 444-458.

[9] M. Steiner, G. Tsudik and M. Waidner, Diffie-Hellman key distribution extended to group communication. In SIGSAC proceedings of the third ACM conference on Computer and Communications Security. New Delhi, India, March 1996., ACM, New York, pp.31-37.

[10] D. Wallner, E. Harder and R.Agee. Key Management for Multicast: Issues and Architectures. Request for Comments (Informational) 2627, Internet Engineering Task Force, June 1999.

[11] C. Wong, M. Gouda, and S. Lam. Secure Group Communication Using key Graphs. In proceedings of the ACM SIGCOMM'98, Oct. 1998.

[12] C.K.Wong., Simon S. Lam., Keystone: A Group key management service, In proceedings of International conference on Telecommunications, Acapulco, Mexico, May 2000.

[13] Safavi-Naini, R., Wang, H.: New Results on Multi-receiver Authentication codes, In Advances in Cryptology-Eurocrypt '98, LNCS, 1438 (1998), 527-541.

[14] Safavi-Naini, R., Wang, H.: Multi-receiver authentication codes: Models, Bounds, Constructions and Extensions, Information and Computation, 151, pp.148-172, (1999).

[15] Safavi-Naini, R., Wang, H.: Broadcast Authentication for Group Communication, Theoretical Computer Science, 269 (1-2), pp.1-21, (2001).

[16] Simmons, G.J.: A Survey of Information Authentication, in Contemporary Cryptology, The Science of Information Integrity, G.J.Simmons, ed., IEEE Press, 1992, 379-419.

**R. Aparna** obtained her M.S. from Birla Institute of Technology, Pilani, Rajasthan, India. She is presently working as an Assistant Professor in the Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India and pursuing Ph.D in the area of Cryptography and Network Security.

**B. B. Amberker** obtained his Ph.D from Department of Computer Science and Automation, IISc., Bangalore, India. He is presently working as Professor in the Department of Computer Science and Engineering, National Institute of Technology, Warangal, AP, India.