Global Journal of Computer Science and Technology
Volume 12 Issue 7 Version 1.0 April 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Heterogeneous Tree Based Authenticated Group Key Transfer Protocol

By A.B.Surekha & C.Shoba Bindu
Jawaharlal Nehru Technological University,Ananatapur

Abstract - Message passing from one source to another has become a key for many upcoming technologies. This is already achieved by introduction of topics of KEYS, AUTHENTICATIONS etc. Secret key transfer is being done presently by using mutually trusted key generation centre (KGS). By this selection of session key by which encryption is done for information passing is selected. This paper discusses about the advancement of this technology by extending this service to group instead of a single key. The whole group with authenticated users can access the information. The proposed protocol considers the heterogeneity of the peer resources as QOS factor in key generation phase and shared key mechanism as primary process to achieve security in group key sharing.

Keywords : GKMP, GKTP, P2P, Group key, QoS, Security.

GJCST Classification: C.2.1

Strictly as per the compliance and regulations of:

Open Association of Research Society, USA
Research | Diversity | Ethics

© 2012. A.B.Surekha & C.Shoba Bindu. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non-commercial use, distribution, and reproduction inany medium, provided the original work is properly cited.

Heterogeneous Tree Based Authenticated Group Key Transfer Protocol

A.B.Surekha [α] & C.Shoba Bindu [σ]

Abstract - Message passing from one source to another has become a key for many upcoming technologies. This is already achieved by introduction of topics of KEYS, AUTHENTICATIONS etc. Secret key transfer is being done presently by using mutually trusted key generation centre (KGS). By this selection of session key by which encryption is done for information passing is selected. This paper discusses about the advancement of this technology by extending this service to group instead of a single key. The whole group with authenticated users can access the information. The proposed protocol considers the heterogeneity of the peer resources as QOS factor in key generation phase and shared key mechanism as primary process to achieve security in group key sharing.

Keywords : GKMP, GKTP, P2P, Group key, QoS, Security.

I. Introduction

Every message under transformation ought to have security provided to it. So, for providing high security, we consider 2 issues namely (1) *Message Confidentiality*: Only the authenticated and intended user should read the message and (2) *Message Authentication*: The receiver should be assured that the sent message is from authenticated sender and the message is not altered in the middle.

Here the work of KGS starts. It should provide a one-time session key to achieve the above 2 issues of key exchange. So, KGS distributes the secret key to all intended users with confidentiality and authentication. We can see from [5] the 2 types of key establishment protocols namely Key transfer protocols, Key agreement protocols.

Apart from this the KGS helps in selecting the secret key and transport them to all communication entities secretly. These session keys are determined by all communication entities where the most commonly used is Diffie-Hellman (DH) key agreement protocol [12].

Public keys of the communication entities play a key role in this protocol. They are exchanged to fix the value of session key. As the public key itself does not provide authentication, uses a digital signature. But the only drawback is that this is on whole applicable only two 2 users but not to a group. The importance of group key is found here as everyone ought to have it. This group key management protocol can be of 2 categories. Centralize group key management protocols, where the whole group is managed by a Group Key generation. Distributed group key management, where each individual manages the generation of key rather than a group key distribution. Of the both key management protocols, we use Centralized group key management the most. It was proposed by Harney et al[15] which takes O(n) where n indicates the size of group participating in the generation of key id. In addition to this, to update this group key either adding or editing the users, we have hierarchical structure based group key protocols [10],[22],[27].

II. Related Work

We have Fiat and Naor[14] introducing a k-resistant protocol. Using this security to about k users is provided with O(k log k log n) keys and server broadcasting $O(k^2 \log^2 k \log n)$ messages per rekeying. EBS (Exclusion Basis System) proposed by Eltoweissy et al.[13] is a combinatorial formulation which helps users to switch between number of keys needed to be stored and number of messages to be transmitted. All this is for key updating so that solution to collusion is provided.

In the previous days, this group generation management protocols involved the naturally generalized DH key agreement protocol. Many examples can be quoted like Ingemarsson et al. [18], Steer et al. [28], Burmester and Desmedt [9],and Steiner et al. [29] . Later, in 1990s, Steiner et al[29] came forward with extension of DH naming it as DH key exchange[29] and in 2001, name was changed to authentication services[6].

Later from 2006, there was a drastic advancement in this group key generations. In the very year of 2006, Bohli[8] proposed a framework for group key generation agreement which is intended to provide security opposing harming participators and active unauthenticated users at every point in the network. In 2007, Katz and Yung [19] proposed the first constant-round and fully scalable group DH protocol which is provably secure in the standard model. Above all, the key feature of group DH is to generate a secret group key by a standardised group like KGS other than relying on members inside.

Author α : M.Tech.,Department of CSE, JNTUACE, Anantapur,INDIA. Email : rekha.pyngas77@gmail.com
Author σ : Associate Professor, Department of CSE, JNTUACE, Anantapur,INDIA. Email : shobabindhu@gmail.com

© 2012 Global Journals Inc. (US)

The next advancement in providing security is identifying the intruders present inside the network. For that, Tzeng [31] provided a conference key agreement protocol with the assistance of discrete logarithm (DL). Each user in the group requires having nm power polynomials with n representing number of participants. Later,. in 2008, Cheng and Lain [11] modified Tseng's conference key agreement protocol based on bilinear pairing. In2009, Huang et al. [16] proposed a no interactive protocol based on DL assumption to improve the efficiency of Tseng's protocol.

All the proposals made and developed till now are good. But one main problem is the time constraint. Since this key agreement involves all the communication entities, takes a lot of time for decision. So to reduce this, we have 2 different solutions. (1)All the communication entities assuming that there is an offline server active all the time and decides the secret key with this assumption.[4],[14],[25,][3]. (2) All the communication entities assuming that an online server is in active state.

Of the two, the 1st one is called key redistribution scheme. In this schema, offline users are provided with a secret piece of information created by a trusted group .But the backhand of this approach is that every server has to store a lot of secret keys and information. So we came to the 2nd approach [20] . It's working is almost similar to IEEE 802.11i standard [17] . Here, an online server votes for a group key and transmits to every group member.

Even though they employ same methodology, there is a slight difference. Instead of encrypt in the group temporal key(GTK) by Key encryption key(KEK) and individually saying the secret key information to each user, here in this approach, the information of group key is also said to all user so that they can calculate their own secret keys. Lain et.al [20] in 1989 was the 1st to come up with an algorithm in this approach making use of (t, n) .It consists of (k-1) members. We can also provide some papers in [2],[21],[25] with the same principle.

Coming to our paper, we are able to make a solution to this problem by providing confidentiality and authentication. We also came forward separating the insider and outsider attacks.

To achieve all the above, every user should have an account in KGS to access the group key transfer service and in turn to achieve a secret key. So, for all these transformations, we need a secret channel for message passing to all the communication entities. And also to transfer this selected group key, to all insiders of network, we need a separate and secret channel. This group key is confidential and no mathematical calculations are involved here but it is information theoretically secure.

III. Objective

Having a look at its background, we should be acquainted with: Choose two large primes p and q and calculate a public n such that $n = p*q$, which can be referred as quandary of factoring.

Practically resolving the quandary of factoring is difficult. Even though Blakely [1] and Shamir [26] developed a solution for this, it is not so efficient. According to this scheme, a whole secret key is shared among all the communication entities so that each gets a share of t. With more or equal to t shares each can calculate their secret keys. But with less than t, computation is not possible. This is called (t,n) scheme. It in turn consists of 2 algorithms:

a) *Share Generation Algorithm:*

Dealer D first picks a polynomial f(x) of degree (t-1) randomly: $f(x) = a_0 + a_1x+,.....a_{t-1}x^{t-1}$, in which the secret $S = a_0 = f(0)$ and all coefficients $a_{0,}a_{1,.....}a_{x-1}$ are in a finite field $IF_p = GF(p)$ with p elements.

D calculates all shares: $S_i = f(i)(\text{mod}_p)$ for $i = 1,......n$ Then,

D calculates a list of n shares $(S_i, S_2,......S_n)$ and distributes each share S_i to corresponding shareholder P_i privately.

b) *Secret Reconstruction Algorithm:*

This algorithm takes any shares $(S_{i_1},....S_{i_t})$ as input, it can reconstruct the secret s as

$$S = f(0) = \sum_{i \in A} S_i \beta_i$$

$$= \sum_{i \in A} S_i \, (\prod_{j \in A-\{i\}} \frac{x_j}{x_{j-}x_i})(\text{mod } p),$$

$A = \{i_1,....i_t\} \subseteq \{1,2,......n\}$, β_i for $i \in A$ are Lagrange coefficients. This scheme is able to satisfy all the security related issues like

a) Able to calculate the secret key only if t or more than t shares are known.
b) If not more than t shares are known, it is not able to calculate the secret key.
c) Also follows the Shamir's scheme that there are no numerical calculations and all are assume base on the above expressions.

After all this a modular inverse is to be calculated for secret reconstruction process. It is discussed in Euclid algorithm [30].

Coming to objectives the proposed protocol is distributed key generation under the consideration of peer resource heterogeneity and security. In proposed protocol model, KGS undertakes the selection of optimized peers to participate in key generation and authenticates the peer integrity and eligibility to become part of the peer network by receiving group key. At the outset every member should register to the KGS which intern at registration selects peers with optimal resources to participate in key generation and provides those selected peers a confidential matter by which calculation of secret key is done and authenticity state of the every peer expecting to be part of the network. Then the selected peers generates group key and for each correct and authorised peer to receive group key, a checksum is appended with cipher text. All around the encryption algorithm provides this security. The confidentiality is achieved by secret sharing scheme proposed. For security, a general broadcast message is created and sent to all communication entities where its secrecy is maintained theoretically.

Considering heterogeneity of the peer resources in key generation and security is the key factor in our paper. So the primary goal is to provide security. Some important goals formulated are:

Selecting peers for key generation: Selecting peers that are optimized in terms of having resources to participate in key generation.

Fixing the key generation peer group count : The proposed protocol selects set of peers such that all other peers can receive group key from selected peer in hop level.

Key freshness : That is, the key should not be used before so that further problems may not arise.

Key Confidentiality : It is the assurance that the secret information is accessed only by authorized group members.

Key authentication : Providing authentication guarantees that generation and broadcasting of secret group key is done by KGS, a trusted organisation but not by any hackers.

In spite of all these QoS and security issues, we have 2 more threats to be worked on

a) Any hacker in person using the authenticated group user for his works done.
b) Hackers modifying the messages in their way of transfer even before reaching the destination esp. KGS.

IV. Proposed Protocol

To overcome these, the proposed protocol has 3 remedial measures.

a) Initialization of KGS.
b) User registration
c) Selecting optimal peers for Group key generation
d) Group key generation and distribution.

Initialization of KGS : In this primary step, KGS chooses optimal peers to participate in group key generation. Then KGC sends all random primes selected as shared checksums of the optimal peers to all peers participating in key generation. Then the peers selected for key generation compute n from shared checksums sent by KGC. This n is made public as stated in the proposed theory above in this paper.

User Registration: Immediately after the KGS is initialized, it is ready to use and encourages the user registrations. It in turn keeps track of all the registered users and alerts optimal peers about unauthorised peers.

Optimal peer selection for key generation : Since the heterogeneity of the peer computational resources has taken into consideration, our proposed model selects the optimal peers with eligible computational resource for group key generation.

Group key generation and distribution : As the registration phase ends with the user requests to the KGS for authentication, it sends the shared checksums of the optimal peers to all optimal peers along with the credentials of eligible peers to optimal peers selected for group key generation. Then optimal peers randomly select the secret key t of the hop level requested user and send him the message which is unique to him. By this he can access the group key.

All this transformations between the KGS and users is fallows.

Step 1: KGS receives certificates and about computational resources from Group members to initiate the key generation.

Step 2 : As the authentication, KGS responses by sending the broadcast messages to selected peers that are optimal in resources to participate in key generation.

Step3 : As a note of agreement, optimal peers send a random challenge $R_i \in Z_n^*$ to KGS.

Step 4 : KGS sends all random challenges as shared checksums of optimal nodes to all optimal nodes.

Then optimal nodes generates group key k from these shared checksums received from GKS, and generates an interpolated polynomial $f(x)$ with degree t' to pass through $(t+1)$ points, $(0,k)$ and $(x_i, y_i \oplus R_i)$, for $i = 1,2,3,....t'$. Optimal nodes also compute t additional points, OP_i for $i = 1,2,3,....t'$,on $f(x)$ and $auth = h(k, OP_1, OP_2, OP_3, ...OP_{t'})$, where h is a one-way hash function and $OP_1, OP_2, OP_3, ...OP_{t'}$ are optimal peers. Then optimal peers send $(auth, OP_i)$, for $i = 1,....,t'$.

Step 5: Every group member, P_i, after knowing the shared secret, $(x_i, y_i \oplus R_i)$, and other optimal

© 2012 Global Journals Inc. (US)

peers OP_i for $i = 1,...,|OP|$, on $f(x)$ P_i able to compute the polynomial $f(x)$ and recover the group key and then P_i computes hash value from k and OP_i for $i = 1,2,3,....t'$ then compares with $auth$ for validity.

V. Results Analysis

The experiments were conducted by developing simulation model using MXML. We build a simulation network with hops count of 80. The simulation parameters described in table 1. Authentication ensures that the buffer is properly allocated to valid packets. The simulation model aimed to compare "Authenticated Group Key Transfer Protocol Based on Secret Sharing" and proposed HTAGKTP. The performance check of these two protocols carried out against to the threats listed below.

- Rushing attack
- Denial of service
- Tunnelling
- The protection against tunnelling attack is the advantage of the HTAGKTP over AGKTP[32].

Number of nodes Range	80
Dimensions of space	1500 m × 300 m
Nominal radio range	250 m
Source–destination pairs	20
Source data pattern (each)	4 packets/second
Application data payload size	512 bytes/packet
Total application data load range	128 to 512 kbps
Raw physical link bandwidth	2 Mbps
Initial ROUTE REQUEST timeout	2 seconds
Maximum ROUTE REQUEST timeout	40 seconds
Cache size	32 routes
Cache replacement policy	FIFO
Hash length	80 bits
certificate life time	2 sec

Table1: Simulation parameters that we considered for experiments

Proposed protocols	Routing strategy	Protects from Rushing attack	Protects from Denial of service	Protects from Routing table modification	Protects from Tunneling
AGKTP[32]	P2p	Yes	Yes	No	No
HTAGKTP	P2p	Yes	Yes	Yes	Yes

Table 2 : Protocols and their ability to handle different attacks

The metrics to verify the performance of the proposed protocol are

- Data packet delivery ratio : It can be calculated as the ratio between the number of data packets that are sent by the source and the number of data packets that are received by the sink.
- Packet Delivery Fraction: It is the ratio of data packets delivered to the destinations to those generated by the sources. The PDF tells about the performance of a protocol that how successfully the packets have been delivered. Higher the value gives the better results.
- Average End To End Delay : Average end-to-end delay is an average end-to-end delay of data packets. Buffering during route discovery latency, queuing at interface queue, retransmission delays at the MAC and transfer times, may cause this delay. Once the time difference between packets sent and received was recorded, dividing the total time difference over the total number of CBR packets received gave the average end-to-end delay for the received packets. Lower the end to end delay better is the performance of the protocol.
- Packet Loss : It is defined as the difference between the number of packets sent by the source and received by the sink. In our results we have calculated packet loss at network layer as well as MAC layer. The routing protocol forwards the packet to destination if a valid route is known; otherwise it is buffered until a route is available. There are two cases when a packet is dropped: the buffer is full when the packet needs to be buffered and the time exceeds the limit when packet has been buffered. Lower is the packet loss better is the performance of the protocol.
- Routing Overhead : Routing overhead has been calculated at the MAC layer which is defined as the ratio of total number of routing packets to data packets.

Figure 1(a) shows the Packet Delivery Ratio (PDR) for basic P2P, AGKTP[32] and HTAGKTP. Based on these results it is evident that HTAGKTP recovers most of the PDR loss that observed in AGKTP[32] against to basic P2P . The approximate PDR loss recovered by HTAGKTP over AGKTP[32] is 1.5%, which is an average of all pauses. The minimum individual recovery observed is 0.18% and maximum is 2.5%. Figure 1(b) indicates AGKTP[32] advantage over

© 2012 Global Journals Inc. (US)

HTAGKTP in Path optimality. HTAGKTP used average 0.019 hops longer than in AGKTP[32] because of the hop level certification validation process of the HTAGKTP that eliminates nodes with invalidate certificate. Here slight advantage of AGKTP[32] over HTAGKTP can be observable.

The packet delivery fraction (PDF) can be expressed as:

$$P' = \sum_{f=1}^{e} \frac{R_f}{N_f}$$

$$P = \frac{1}{c} * P'$$

- P is the fraction of successfully delivered packets,
- c is the total number of flow or connections,
- f is the unique flow id serving as index,
- R_f is the count of packets received from flow f
- N_f is the count of packets transmitted to flow f .

Figure 1(c) confirms that HTAGKTP is having fewer packets overhead over AGKTP[32]. Due to stable paths with no compromised or victimized nodes determined by HTAGKTP this advantage become possible. The Packet overhead observed in AGKTP[32] is average 5.29% more than packet overhead observed in HTAGKTP. The minimum and maximum packet overhead in AGKTP[32] over HTAGKTP observed is 3.61% and 7.29% respectively.

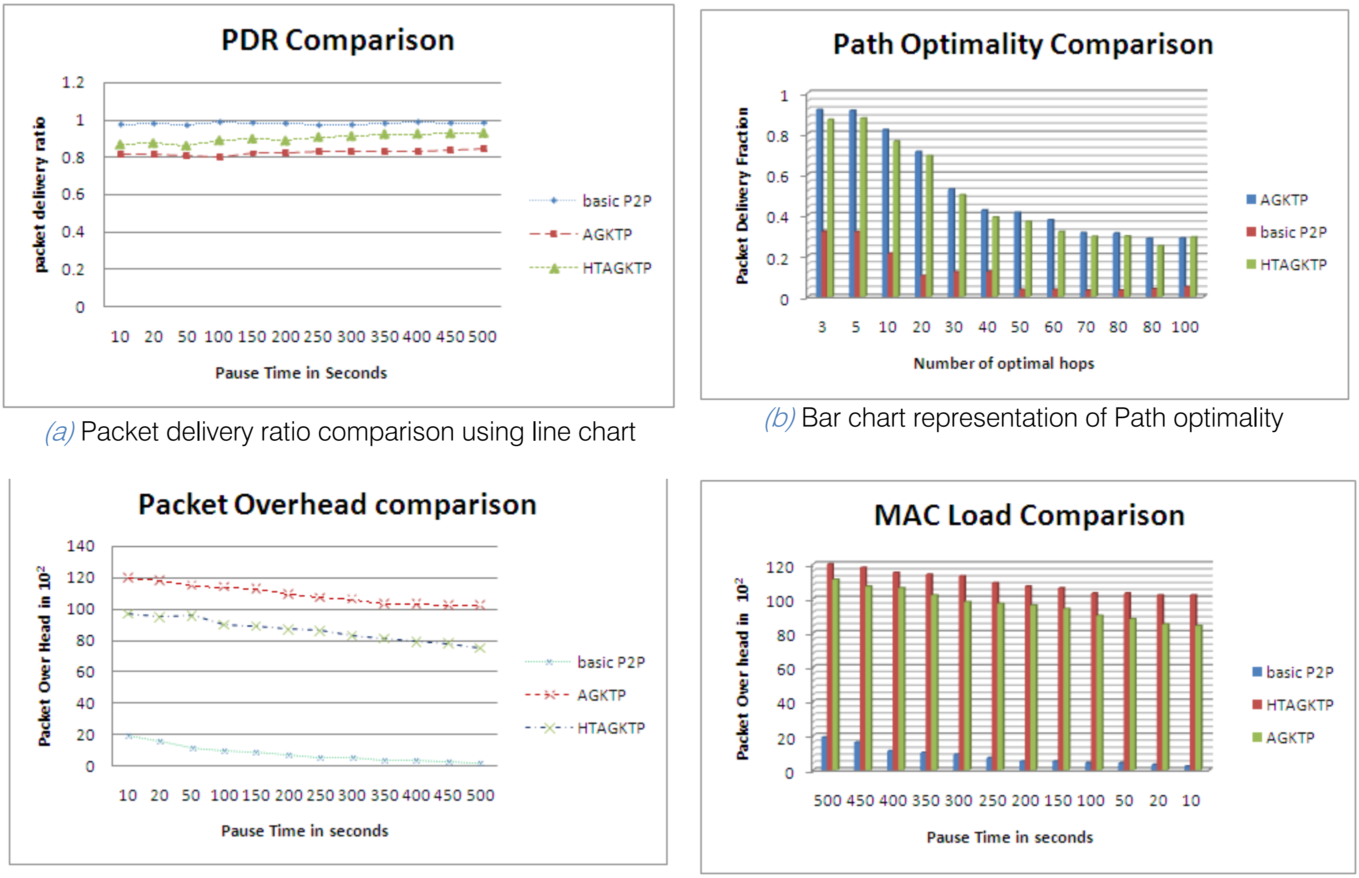
MAC load overhead is slightly more in HTAGKTP over AGKTP[32]. We can observe this in Figure 1(d), which is because of additional control packet exchange in HTAGKTP for neighbour hop validation through certificate exchange. The average MAC load overhead in HTAGKTP over AGKTP[32] 1.64%. The minimum and maximum MAC load overhead observed is 0.81 and 3.24% respectively.

In all these evaluation strategies the results derived for basic P2P are interesting. In all metrics except path optimality, basic P2P performed well since it is not considering any security issue as routing parameter, and it is delivering better QOS under no security threat in routing assumption, which is not true in real time practices. In path optimality validation among three considered protocols basic P2P stands last because it is not considering any security constraints, hence identifies unstable paths.

VI. Conclusion

Tight security mechanisms are needed to allow secure communication among the group members. Thus, a communication session must have security services to provide authentication, integrity, and confidentiality. Group Key (GK) is the primary and key part of the safe group communication. The performance of GK generation process, which is required for secure communication, may degrade due to less performing members. Thus, the generation process must be done is a more precise way but filtering less performing members. Many changes are occurring in the recent years as increase in usage of mobile computers, network clusters communication with standard servers. Apart from this, heterogeneity and distributed computer environment became common in the current internet world. Thus, GK management system must consider various parameters, differences and environments involved in the communication. These considerations as the basis, the effectiveness of HTAGKTP protocol in comparison to AGKTP[32] is proved. This protocol improves the efficiency by considering the parameters effecting the performance i.e. computational delay and network latency. Thus, this research is aimed at and thus proved that GKGP is more efficient and maximizes the applicability of communication.

© 2012 Global Journals Inc. (US)



(a) Packet delivery ratio comparison using line chart

(b) Bar chart representation of Path optimality

(c) A line chart representation of Packet overhead comparison report

(d) Mac load comparison represented in bar chart format

Figure 1: Evaluation report of HTAGKTP performance over AGKTP[32]

References Références Referencias

1. G.R. Blakley, “Safeguarding Cryptographic Keys,” Proc. Am. Federation of Information Processing Soc. (AFIPS ’79) Nat’l Computer Conf., vol. 48, pp. 313- 317, 1979.
2. S. Berkovits, “How to Broadcast a Secret,” Proc. Eurocrypt ’91 Workshop Advances in Cryptology, pp. 536-541, 1991.
3. R. Blom, “An Optimal Class of Symmetric Key Generation Systems,” Proc. Eurocrypt ’84 Workshop Advances in Cryptology, pp. 335-338, 1984.
4. C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly Secure Key Distribution for Dynamic Conferences,” Information and Computation, vol. 146, no. 1, pp. 1-23, Oct. 1998.
5. C. Boyd, “On Key Agreement and Conference Key Agreement,” Proc. Second Australasian Conf. Information Security and Privacy (ACISP ’97), pp. 294-302, 1997.
6. E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, “Provably Authenticated Group Diffie-Hellman Key Exchange,” Proc. ACM Conf. Computer and Comm. Security (CCS ’01), pp. 255-264, 2001.
7. J.M. Bohli, “A Framework for Robust Group Key Agreement,” Proc. Int’l Conf. Computational Science and Applications (ICCSA ’06), pp. 355-364, 2006.
8. M. Burmester and Y.G. Desmedt, “A Secure and Efficient Conference Key Distribution System,” Proc. Eurocrypt ’94 Workshop Advances in Cryptology, pp. 275-286, 1994.
9. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, “Multicast Security: A Taxonomy and Some Efficient Constructions,” Proc. IEEE INFOCOM ’99, vol. 2, pp. 708-716, 1999.
10. J.C. Cheng and C.S. Laih, “Conference Key Agreement Protocol with Non Interactive Fault-Tolerance Over Broadcast Network,” Int’l J. Information Security, vol. 8, no. 1, pp. 37-48, 2009.
11. W. Diffie and M.E. Hellman, “New Directions in Cryptography,” IEEE Trans. Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.
12. M. Eltoweissy, M.H. Heydari, L. Morales, and I.H. Sudborough, “Combinatorial Optimization of Group Key Management,” J. Network and Systems Management, vol. 12, no. 1, pp. 33-50, 2004.
13. A. Fiat and M. Naor, “Broadcast Encryption,” Proc. 13th Ann. Int’l Cryptology Conf. Advances in Cryptology (Crypto ’93), pp. 480-491, 1994.

14. H. Harney, C. Muckenhirn, and T. Rivers, “Group Key Management Protocol (GKMP) Architecture,” RFC 2094, July 1997.
15. K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai, and T.S. Chen, “A Conference Key Agreement Protocol with Fault-Tolerant Capability,” Computer Standards and Interfaces, vol. 31, pp. 401-405, Jan. 2009.
16. IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.
17. I. Ingemarsson, D.T. Tang, and C.K. Wong, “A Conference Key Distribution System,” IEEE Trans. Information Theory, vol. IT-28, no. 5, pp. 714-720, Sept. 1982.
18. J. Katz and M. Yung, “Scalable Protocols for Authenticated Group Key Exchange,” J. Cryptology, vol. 20, pp. 85-113, 2007.
19. C. Lain, J. Lee, and L. Harn, “A New Threshold Scheme and Its Application in Designing the Conference Key Distribution Cryptosystem,” Information Processing Letters, vol. 32, pp. 95-99, 1989.
20. C.H. Li and J. Pieprzyk, “Conference Key Agreement from Secret Sharing,” Proc. Fourth Australasian Conf. Information Security and Privacy (ACISP ’99), pp. 64-76, 1999.
21. A. Perrig, D. Song, and J.D. Tygar, “Elk, A New Protocol for Efficient Large- Group Key Distribution,” Proc. IEEE Symp. Security and Privacy, pp. 247-262, 2001.
22. G. Saze, “Generation of Key Predistribution Schemes Using Secret Sharing Schemes,” Discrete Applied Math., vol. 128, pp. 239-249, 2003.
23. A. Shamir, “How to Share a Secret,” Comm. ACM, vol. 22, no. 11, pp. 612- 613, 1979.
24. A.T. Sherman and D.A. McGrew, “Key Establishment in Large Dynamic Groups Using One-Way Function Trees,” IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.
25. D.G. Steer, L. Strawczynski, W. Diffie, and M.J. Wiener, “A Secure Audio Teleconference System,” Proc. Eighth Ann. Int’l Cryptology Conf. Advances in Cryptology (Crypto ’88), pp. 520-528, 1988.
26. M. Steiner, G. Tsudik, and M. Waidner, “Diffie-Hellman Key Distribution Extended to Group Communication,” Proc. Third ACM Conf. Computer and Comm. Security (CCS ’96), pp. 31-37, 1996.
27. D.R. Stinson, Cryptography Theory and Practice, second ed., CRC Press, 2002.
28. W.G. Tzeng, “A Secure Fault-Tolerant Conference Key Agreement Protocol,” IEEE Trans. Computers, vol. 51, no. 4, pp. 373-379, Apr. 2002.
29. Harn, L.; Changlu Lin; , "Authenticated Group Key Transfer Protocol Based on Secret Sharing," Computers, IEEE Transactions on , vol.59, no.6, pp.842-846, June 2010, doi: 10.1109/TC.2010.40

© 2012 Global Journals Inc. (US)

This page is intentionally left blank

© 2012 Global Journals Inc. (US)