Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

A NOVEL CRYPTOGRAPHIC METHOD FOR DATA NETWORKS' DYNAMIC CONFERENCE SECURITY

Mr. Bijay Kumar Sahoo[1], Mr. Sunil Kumar Mishra[2]*

[1] Assistant Professor Dept. Of Computer Science and Engineering, NIT , BBSR*
[2]Assistant Professor,Dept. Of Computer Science and Engineering, NIT , BBSR
bijaykumar@thenalanda.com sunilmishra@thenalanda.com*

***Abstract*— Dynamic conferencing describes a situation in which any subset of users in a user universe convene in a conference to exchange private information. Computing a shared secret key for such a dynamically constructed conference is known as the key distribution (KD) problem in dynamic conferencing. The KD systems for dynamic conferencing described in the literature either require communication between users, which is undesirable, or are computationally impractical. One such KD approach, called the extended symmetric polynomial based dynamic conferencing scheme (ESPDCS), has a high computational complexity that depends on the size of the universe. We provide an improvement to the ESPDCS scheme in this research to create a universe- independent SPDCS (UI-SPDCS) KD technique, whose complexity is independent of the size of the universe. The UI-SPDCS scheme, however, does not grow with the size of the conference. Using the help of the UI-SPDCS scheme and the tree-based group Diffie- Hellman (TGDH) key exchange protocol, we suggest the KD scheme known as DH-SPDCS, which is reasonably scalable. The suggested DH-SPDCS technique offers a programmable trade-off between the complexity of the computation and communication.**

I. INTRODUCTION

Research on secure communications has mostly focused on two-party and recently on group communications (multi-party). In a secure group communication (SGC) setting, a group of two or more users share information secretly such that any user not in the group cannot glean or manipulate any information being shared. Achieving security in group communications is more difficult than in point-to-point communications. Various schemes have been proposed for key distribution in secure group communications, such as key-tree based schemes, tree-based group Diffie-Hellman (TGDH), Iolus, dual encryption protocol (DEP), DIstributed Scalable sEcure Communication

(DISEC), and a suite of *n*-party Diffie-Hellman keyexchange schemes (refer to [1] for details).

The universe of users is represented by set U of size n. Any set of users from U can form a dynamic conference to share confidential information among themselves. The number of potential dynamic conferences possible in a universe of n users is $(2^n \ (n + 1))$, which is exponential in number. A user can participate in more than one conference at the same_time. All or any number of dynamic conferences can co-exist in the system at the same time. A burst of users can join or leave a conference. Following the join/leave operations, the conference key needs to be updated to maintain backward/forward secrecy. The size of the dynamic conference can vary in the range [2 n]. In most cases the conference size c is $c << n$.

Let us discuss the inherent complexity of the key distribution/computation (KD) problem

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

in dynamic conferencing (DC). Each user can potentially belong to half the number of conferences. Evidently, scalability is one of the most important issues for the KD problem in DC. A key distribution scheme (KDS) for DC should allow a user to compute shared secret keys for all the conferences that he/she belongs to. At the same time, the scheme should prevent a user, having the knowledge of conference keys of all the conferences he/she belongs, from computing shared secret keys of the other conferences he/she does not belong to. Such a dynamic conferencing scheme is said to be 1- secure. Similarly, in a *w*-secure key distribution scheme for dynamic conferencing even if *w* users collude and pool together all their information (shared secret keys or any other) they should not be able to compute secret keys of any of the conferences that they do not belong to. Other important features of a KD scheme for DC include maintaining forward backward secrecy, efficient handling of bursty join/leave operations, and minimizing the interaction among users for shared secret key computation.

The rest of the paper is organized as follows. Section II discusses the available schemes in the literature for dynamic conferencing and our motivation of our work. In Section III we present KD scheme called UI-SPDCS whose computational complexity is independent of the size of the universe. In Section IV we discuss the TGDH scheme and present our DH-SPDCS scheme. In Section V we present a security analysis of the proposed DH-SPDCS scheme. Section VI discusses the results of simulation of DH-SPDCS for dynamic conferencing. Section VII concludes the paper.

II. RELATED WORK AND MOTIVATION

In this section we survey a few key distribution schemes available in the literature for dynamic conferencing (DC) and describe their advantages and disadvantages. All the KDS schemes can be classified into two main classes. The Class 1 KDS schemes [11]-[15] are used for securing large dynamic conferences but they require interaction among the users in the dynamic conference for computing the common secret key of the conference. The second class of KDS schemes are used for securing small and medium scale dynamic conferences but they require no communication among the users to compute the shared conference key.

The Class 1 KDS schemes include, the interval based scheme [7], the unconditionally secure dynamic conference scheme [6], asymmetric encryption algorithms using public key encryptions [5], the secure lock scheme [5]. These scheme have issues such In [2], Zou et al., proposed an extension to the SPDCS scheme called ESPDCS where any group of $t \leq n$ users from a universe can compute the shared secret key. The computational complexity of ESPDCS is $O((w + 1)^n)$, which is clearly exponential in the size of the universe n. In the next section we propose a KD scheme called universe independent symmetric polynomial based dynamic conferencing scheme (UI- SPDCS) based on the ESPDCS scheme to reduce its complexity from $O((w + 1)^n)$ to $O((w + 1)^k)$ where k is the maximum size of any conference in a universe U of users.

III. THE UNIVERSE-INDEPENDENT SPDCS(UI-SPDCS)

The universe-independent symmetric polynomial based dynamic conferencing scheme (UI-SPDCS) is a generalization of the ESPDCS scheme. Let us present the UI-SPDCS key distribution scheme for a universe $U = \{u_1, u_2, \ldots u_n\}$ of size n with security parameter $w \leq n$

2. Let q be a prime integer such that $q \geq n$ and GF(q) be the Galois Field of q. To setup a key distribution scheme the central trusted authority (CTA) chooses $n + k$ random numbers $S = \{s_1, s_2, \ldots s_n\}$ and $S' = \{s'_1, s'_2, \ldots s'_k\}$ in GF(q) and a symmetric polynomial $f(x_1, \ldots, x_k)$ in k variables and degree w in each variable with coefficients in $GF(q)$.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

The CTA is notone of the users in U and is only active during the setupphase of the key distribution scheme. The S and S' aremade public and the symmetric polynomial is kept secret and is known only to the CTA.

A polynomial $f(x_1,\ldots, x_k)$ (see Eqn.1) is symmetric if and only if $a_{i1,\ldots,ik} = a_{\pi(i1),\ldots,\pi(ik)}$ for all monomials as lack of scalability, inefficient key computation, no $a_{i,\ldots,i}$ $x^{i1},\ldots, x^{ik}$ in f and all permutations π of

$$1 \quad n \quad 1 \quad k$$

support for join/leave operations, etc.

Blom in [8] presented a symmetric key generationsystem for two-party communication using (n, k) MDS $1,\ldots, k$.

$$f(x_1,\ldots x_k) = \sum \cdots \sum a_{i,\ldots,i}\, x^{i1} \cdots x^{ik}. \qquad (1)$$

$$w \qquad w$$

linear codes where n is the universe size and k isthe security parameter. The amount of information $i1 =0$

$$1 \quad k \quad 1 \quad k$$

$i_k=0$

stored by each user is bounded by $(k \log q)$ bits. In [9], Matsumoto et al., extended Blom's MDS based scheme to general symmetric t-linear mappings (GSM) for dynamic conferences of size t in a universe of n users. Blundo et al., in [4], proposed a symmetric polynomial based dynamic conferencing scheme (SPDCS) for dynamic conferences of size t. The computational complexity of the SPDCS scheme is $O((w + 1)^t)$, which is pseudo polynomial assuming that t is an arbitrary constant that specifies the size of the conference. One ofthe disadvantages of this scheme is that for a particular key distribution scheme exactly t users have to form a secure group: no more, no less. Therefore we need a different scheme for every choice of t users in a group.

Therefore, if $f(x, y, z)$ is a symmetric polynomial then $f(x, y, z) = f(y, z, x) = f(z, x, y)$. For example $f(x, y, z) = x + y + z + 2xy + 2yz + 2zx$ is a symmetric polynomial. The secret polynomial f consists of $(w + 1)^k$ terms. For $1 \leq i \leq n$, the CTA computes a polynomial g_i in $k - 1$ variables $g_i = f(x_1 = s_i, x_2,\ldots, x_k)$ by substituting $x_1 = s_i$ and sends it secretly to user u_i. The resulting polynomial g_i is also symmetric and contains at most $(w + 1)^{n-1}$ terms. The coefficients of g_i comprise the secret information share which is given to user u_i. Each user u_i is assigned public values s_i and private share g_i.

Now, any subset of members of size $t \leq k$ can compute a shared secret key. Let us suppose that

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

users in the set $\{u_1, u_2, \ldots u_t\}$ are forming a dynamic conference. The conference key is calculated by each participating user u_i by substituting the public S values of each of the other users in the conference in place of variable x_j's in the secret polynomial of g_i of user u_i. The remaining $k - t$ variables in the polynomial g_i are substituted by the first $(k-t)$ S' values. Since polynomial f is symmetric $f(s_1, s_2, \ldots, s_t, s'_1, \ldots, s'_{k-t}) = g_i(s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_t, s'_1, \ldots, s'_{k-t})$ for all $1 \leq i \leq t$. Though, in the above example we have considered the conference with the first t users $s_1, s_2, \ldots, s_t$ of the universe the scheme holds for any set of at most k users. The UI-SPDCS scheme has all the advantages of the ESPDCS scheme and its complexity is only $((w + 1)^k)$, independent of the size of the universe. The scheme is w-secure where collusion of any k members $k \leq w$, cannot break the system security. The security of the UI-SPDCS scheme follows directly from the security of the basic SPDCS scheme.

A user u_i can compute a secret key for himself by substituting x_j with s'_j in his secret polynomial g_i for all $j \leq i$. Since no other user knows the polynomial g_i the secret key for u_i cannot be computed by any other user
in the universe. This self secret key computation is used in the DH-SPDCS scheme, discussed later in Section IV. The computational complexity of the UI-SPDCS scheme is $O((w+1)^k)$ which is still exponential in the size of the conference. In the next section we present a KD scheme that uses both the UI-SPDCS scheme and the Tree-based Group Diffie-Hellman (TGDH) key exchange protocol to further scale the size of the conference.

IV. DIFFIE-HELLMAN AND UI-SPDCS BASED DYNAMIC CONFERENCING SCHEME (DH-SPDCS)

In this section we present our KD scheme for dynamic conferencing that uses the UI-SPDCS scheme and the TGDH protocol. The DH-SPDCS key distribution scheme for dynamic conferencing in a universe of n users is implemented in three phases, namely *universe partitioning phase*, *setup phase* and *key computation phase*. We now discuss each of the three phases of the key computation time of dynamic conferencing within a partition. The partitioning of the universe can be uniform where all the m partitions have equal number of users (with the exception that the last partition may have more/less number of users if n is not a multiple of m). The universe can also be partitioned non-uniformly based on various criteria such as temporal proximity, frequent dynamic conferencing, adversarial collaboration etc. Members of an organization conferencing more frequently can be grouped into a single partition. The members in an organization that are likely to collaborate can be partitioned into different groups. We will describe in the later sections that the security of the proposed DH-SPDCS scheme is only vulnerable to users colluding within the same partition.

B. Setup Phase

In the setup phase, the CTA does the initialization of the UI-SPDCS scheme for each partition in the universe independently. For each partition T_i, the CTA considers the users in partition T_i as a universe in itself and initializes the UI-SPDCS scheme Ψ_i. The security parameter of each partition T_i is configured depending upon the number of users in the partition and specific security requirements of the partition. Let w_i, $w_i \leq n_i - 1$, be the security parameter of the partition T_i. Most of the overhead involved with UI-SPDCS is during the setup phase that is done offline. The CTA also computes the Diffie-Hellman key tree

and gives it to each user in the conference. All the users belonging to one partition are associated to one leaf node in the DH key tree. The number of leaf nodes of the DH key tree is equal to the number of partitions of the universe.

Algorithm 1 COMPUTE-KEY(U, Λ, S)

1: **if** (S T_i) for some T_i Λ **then** ⊆ ∈
2: Compute shared secret key K_S of S using SPDCS Ψ_i of partition T_i
3: **else**
4: **for** $T_i \in \Lambda$ **do**
5: Compute $S_i = S \cap T_i$
6: **if** S_i /= φ **then**

DH-SPDCS KD scheme in detail.

A. Universe Partitioning Phase

The universe U of n users is partitioned into m virtualpartitions $\Lambda = T_1, T_2, \ldots T_m$ such that each user in U belongs to exactly one virtual partition. Formally, 7: Compute shared secret key K_{Si}

SPDCS Ψ_i of partition T_i
8: **else**
9: Let $K_{Si} = 1$
10: **end if**
11: **end for**of S_i using
for all i, j such that i j, $T_i \cap T_j = \varphi$ and
12: Using ($K_{S1}, K_{S2}, \ldots K_{Sm}$) as secret keys of the $\bigcup_{i=1}^{m}$ i=1
$T_i = U$. The number of users in virtual partition
m leaf nodes in the DH key tree of TGDH, compute root key
T_i is represented by n_i. The maximum number of users in a partition can be bounded by a thresholdvalue depending upon the bounded requirement on the
13: **end if**

C. Key Computation Phase

In this section we will discuss how to compute the shared secret key of a subset S of users from the universe U. Algorithm 1 outlines the conference key computation using DH-SPDCS. If all the users in the subset S belong to one single virtual partition T_i i.e., S T_i the secret key K_S of the conference is computed using UI-SPDCSof the partition T_i. The key K_S is computed without any interaction between users in the dynamic conference S. We call such a conference key as *intra partition key*. By limiting the maximum number of users in a partition, the intra partition key computation time is bounded.

Let a dynamic conference S span across multiple partitions. The shared secret key of such a dynamic conference is called *inter partition key*. Let S_i be the set of users in the dynamic

conference S that belong to partition T_i. The users in each set S_i corresponding to partition T_i compute their intra partition key K_{Si} independently and in parallel using their corresponding UI-SPDCS scheme Φ_i. If $S_i = \varphi$ then intra partition key of S_i is 0 and the corresponding public key is 1. If $S_i = 1$, the user can compute a secret key for himself which no other user/s can compute. Nodes in each partition knowing its private share and the DH key tree can compute the shared secret of the dynamic conference S using the TGDH scheme.

The tree-based group Diffie-Hellman (TGDH) [10] is a protocol to compute a shared secret key for secure group communication. The TGDH key distribution protocol specifies how a user, knowing only his secret key and a binary key tree, computes the shared secret key of the group. The advantage of the TGDH scheme is that there is no central authority. The binary key tree is known to every user in the group and is called the Diffie-Hellman (DH) key tree. The TGDH scheme also specifies how to recompute the shared secret key after user join/leave operations.

D. *Bursty Join/Leave Events*

In this section we discuss the key re-computation when one or more users join or leave a conference. Following the bursty join/leave event, the secret share of the members of each partition needs to be re- computed. Depending upon the number of members from each partition joining or leaving the conference we have four different cases. The initial conference as well as the resulting conference can be a conference with all the members belonging to a single partition or members belong to more than one partition. In any of the cases, the conference key updating is same as the conference key computation assuming that the resulting conference is a new conference. If in the resulting conference, members belong to a single partition then the conference key is computed using the UI-SPDCS key distribution scheme corresponding to that partition. But if the resulting conference has members spanning across multiple partitions then the conference keys of the members belonging to each partition are recomputed in parallel which correspond to the secret keys of the leaf nodes in the DH key tree. After updating the key shares of the leaf nodes in the DH key tree the new shared secret key is computed using the TGDH key exchange protocol.

E. *Evaluation*

Table IV-C summarizes the important features of dynamic conferencing that the DH-SPDCS scheme supports, in comparison with other schemes. DH-SPDCS allows distributed key computation (with little or no interaction), efficient join/leave and is computationally secure. It is also more scalable in comparison to other schemes. DH-SPDCS thus proves to be a major improvement from its earlier counterpart namely ESPDCS and also other available schemes. The UI- SPDCS scheme is an improvement to the ESPDCS scheme in terms of the universe size scalability.

V. ANALYSIS OF SECURITY OF DH-SPDCS

In this section we present a preliminary analysis of the security of the DH-SPDCS scheme.

A. *Security of DH-SPDCS*

If w_i is the security parameter of the UI-SPDCS KDS for the partition T_i then $w_i + 1$ users inside the partition T_i need to collude to break the security of the UI-SPDCS. But the security of the UI-SPDCS of a partition T_i cannot be compromised by the collusion of any number of users from other partitions T_j where $i = j$. The security parameter of the DH-SPDCS scheme for all the

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

dynamicconferences within a partition T_i is given by a tuplew_l, w_g where w_l represents the threshold for security against the maximum number of colluding users insidethe partition and w_g represents the threshold for securityagainst the maximum number of colluding users outsidethe partition T_i. For this scheme $w_l = w_i$ and $w_g =$ but the maximum number of users in universe U outsidethe partition T_i is n n_i. Hence $w_g = (n \quad n_i \quad 1)$.

Let us now consider the security of the DH-SPDCS scheme for dynamic conferences that span more than one partition in Λ. Let S be a dynamic conference that spans m partitions $T_S = \{T_{i1}, T_{i2}, \ldots T_{im}\}$ in Λ of the universe U. The security parameter of the DH-SPDCS scheme for dynamic conference S spanning m partitions T_S Λ, is given by a $(m + 1)$-tuple $\langle w_{i1}, w_{i2}, \ldots w_{im}, w_g \rangle$ where w_{ij} represents threshold for security against the maximum number of colluding users inside the partition T_{ij} and w_g represents the threshold for security against the maximum number of colluding users outside the partitions T_S involved in the conference S. To break the security of such a system one has to break the security of UI-SPDCS of at least one ofthe partitions T_S in Λ which have at least one user fromthe conference S. We know that if w_k is the security parameter of the UI-SPDCS for key distribution in thepartition T_k then w_k +1 users need to collude to break thesecurity of the UI-SPDCS. For the dynamic conference

	Universe Size Independence	Scalability	No Communication	Distributed Key	Join/Leave	Computational Security
Uncond. Secure	X	X	C	X	X	C
Public Key Based	C	X	X	X	C	X
Interval Based	X	X	X	X	C	C
Secure Lock	C	X	X	X	C	X
ESPDCS	X	X	C	X	C	C
UI-SPDCS	C	X	C	X	C	C
DH-SPDCS	C	C*	X	C	C	C

TABLE I
DYNAMIC CONFERENCING SCHEMES AND PROPERTIES. C: PARTIALLY SCALABLE.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

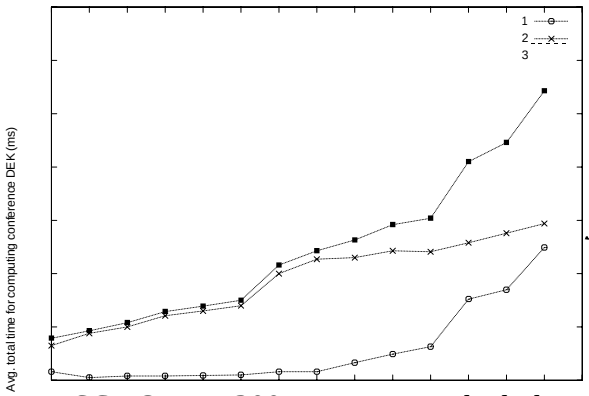
all the users in the conference to compute the root key using the TGDH protocol as $time_{rk}$. The root key is the conference key which only the users in the conference can compute. The average total time required to compute the conference key by all the users in the conference is $time_{ck} = time_{igk} + time_{rk}$.

700

600

$S, w_g = n - \sum_{i=1}^{m} n_i$.

500



VI. RESULTS AND DISCUSSION

We implemented SPDCS and DH-SPDCS on an SGI Origin 300 computer, which has 32 processors and 16 GB of RAM. The machine has a 5^{th} minute load average[1] of around 40%. The scheme was implemented in C++ along with the GMP (GNU's Multiple Precision) library and POSIX thread libraries. Below we discuss and compare the results of DH-SPDCS with the results of SPDCS. The simulation is run for 5 sets of 30 random events with

400

300

200

100

0

12 18 24 30 36 42 48 54 60 66 72 78 84 90 96

Total no. of clients in universe

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 03, March : 2022

security parameter $w = 1$.

A. Results

During a simulation we assume that universe size is constant. In all our simulations, we uniformly partition the universe into $m = 6$ partitions and hence the tree structure of DH-SPDCS remains the same. The size of the universe varies in the range 12 to 90 and correspondingly the number of users in each partition n in the range 2 to 15.

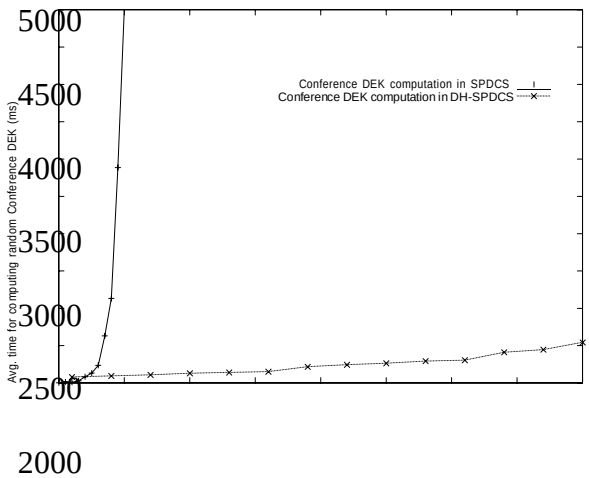
We use the time taken to compute the conference keyas a measure of performance. The average time taken by all the members in a conference to compute the intra partition key using the UI-SPDCS scheme is referredto as $time_{igk}$. Let us term the average time taken by

[1]5th minute load average: This values indicates the average number of processes competing for getting scheduled on one of the CPUs during the 5 minutes while simulation was running on the machine.

Fig. 1. Conference key computation time for a single member;
Description of curves: 1. $time_{igk}$, 2. $time_{rk}$, and 3. $time_{ck}$

Figure 1 plots all the three curves namely $time_{igk}$, $time_{rk}$, and $time_{ck}$ are referred to as curve 1, 2 and 3 respectively. Assuming that users do not collude, the security parameter for all the UI-SPDCS schemes of all the partitions is set to 1. Though the x-axis shows the number of users in the universe varying from 12 to 96 the number of users in each partition of the universe varies from 2 to 15. Therefore at $x = 90$, the curve 1 shows that the time taken by UI-SPDCS for computing intra partition key with only 15 users in the partition averaged over all the 6 partitions of the universe. The curve 1 shows that $time_{igk}$ increases exponentially with n. It has also been observed that for values of n greater than 15 (not shown in the graph) the key computation time increases beyond practical feasibility and is not suitable for real world applications. Hence during the simulationof DH-SPDCS we limit the value of n to 15. The curve

2 in Figure 1 plots the time taken to compute the root key using the TGDH protocol ($time_{rk}$) having computed the secret of the leaf nodes of the DH key tree using UI-SPDCS. The curve 2 shows that the time $time_{rk}$ curve is linear and increases very slowly with the increase in the number of users in a partition. And finally the curve 3 plots the average total time taken to compute the conference key by all the users in the conference using DH-SPDCS ($time_{ck}$).



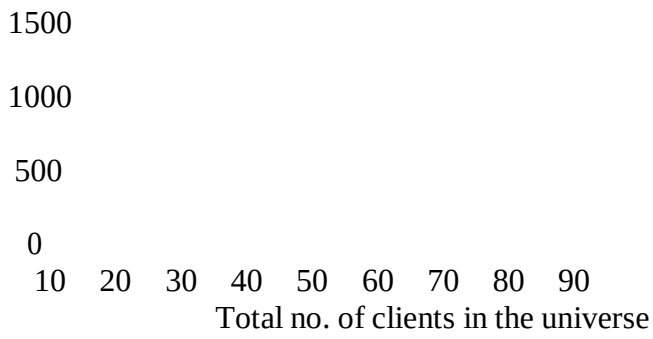


Fig. 2. Comparison between SPDCS and DH-SPDCS.

Figure 2 shows the time taken to compute the conference key for the SPDCS and DH-SPDCS schemes. The graph clearly shows that when the number of users in the universe increases to 20 the time taken by SPDCS to compute the conference key increases to more than 5, 000*ms*. In contrast the time taken by our proposed DH-SPDCS even when the size of the universe is 90 is just over 500*ms*. This clearly substantiates the huge improvement in scalability of DH-SPDCS scheme in comparison to SPDCS.

VII. CONCLUSION

In this paper we studied the key distribution problem for dynamic conferencing in a universe of users. The ESPDCS scheme is a key distribution scheme for dynamic conferencing that allows us to compute the secret keys for dynamic conferences without user interaction. In this work we presented an enhancement to the ESPDCS scheme such that it is scalable with the size of the universe – the universe independent SPDCS (UI-SPDCS) scheme. In this work we proposed a KD scheme for dynamic conferencing termed as DH-SPDCS that uses the UI-SPDCS scheme and the tree-based group Diffie-Hellman (TGDH) protocol. We presented an analysis of the security of the DH-SPDCS scheme. We simulated DH-SPDCS and found that it increases the range of the ESPDCS scheme. However, the scalability
is attained at the expense of increased computational complexity in shared key computation.

VIII. ACKNOWLEDGEMENTS

This work is partially supported by NSF Grant No.
CCR-0311577.

REFERENCES

[1] X. Zou, B. Ramamurthy, S. S. Magliveras, "Secure Group Communications Over Data Networks," Boston, MA: Springer, ISBN: 0-387-22970-1, October 2004.

[2] X. Zou, S. Magliveras, and B. Ramamurthy, "A dynamic conference scheme extension with efficient burst operation," *Congressus Numerantium*, 158:83–92, 2002.

[3] C. Blundo, et al., "Generalized Beimel–Chor schemes for broadcast encryption and interactive key distribution," *Theoretical Computer Science*, 200(1–2):313–334, 1998.

[4] C. Blundo, et at., "Perfect secure key distribution for dynamic conferences," *Advances in Cryptology (CRYPTO '92)*, pages 471–486, November 1993.

[5] G. H. Chiou and W. T. Chen, "Secure broadcasting using the secure lock," *IEEE Transactions on Software Engineering*, 15(8):929– 934, August 1989.
[6] Y. Desmedt and V. Viswanathan, "Unconditionally secure dynamic conference key distribution," *Proceedings of the IEEE International Symposiun on Information Theory, NY, USA*, pages 383–383, August 1998.
[7] M. G. Gouda, et al., "Key trees and the security of interval multicast," *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, 2002.
[8] R. Blom, "An optimal class of symmetric key generation systems," *Advances in Cryptology - EUROCRYPT'84, LNCS, Springer, Berlin*, vol. 209, pp. 335-338, 1985.
[9] T. Matsumoto and H. Imai, "On the Key Pre-distribution System: A Practical Solution to the Key Distribution Problem," Advances in Cryptology: Proceedings of Crypto 87, Lecture Notes in Computer Science, vol. 239, Springer - Verlag, Berlin, 1987, pp. 185-193.
[10] Y. Kim and A. Perrig and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," Seventh ACM Conference on Computer and Communications Security, CCS-7, pp. 235-244, 2000.
[11] S. S. Kulkarni and B. Bezawada, "Distributing Key Updates in Secure Dynamic Groups," in Proceedings of the International Conference on Distributed Computing and Internet Technology, ICDCIT 04, Bhubaneswar, India.
[12] X. Zou, S. S. Magliveras, and B. Ramamurthy, "Key Tree based Scalable Secure Dynamic Conferencing Schemes," in proceedings of International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), MIT Cambridge, MA, USA, November 2004.
[13] D. A. McGrew and A. T. Shermany, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," *Transactions on Software Engineering*, vol. 29, no. 5, pp. 444-458, May 2003.
[14] P. Adusumilli and X. Zou, "KTDCKM-SDC: A Distributed Conference Key Management Scheme for Secure Dynamic Conferencing," in the proceedings of the tenth IEEE Symposium on Computers and Communications (ISCC), Cartagena, Spain, June 27-30, 2005, pp. 476–481.
[15] G. Caronniy, M. Waldvogelz, D. Sunz, B. Plattnerz, "Efficient Security for Large and Dynamic Multicast Groups," in the proceedings of the Seventh Workshop on Enabling Technologies, (WETICE 98), IEEE Computer Society Press, 1998.