A novel Conference Key Management solution for Secure Dynamic Conferencing

Xukai Zou*

Department of Computer and Information Science,
Indiana University-Purdue University Indianapolis,
IN 46202, USA
E-mail: xkzou@cs.iupui.edu
*Corresponding author

Yogesh Karandikar

2Wire Inc., California 95131, USA
E-mail: ykarandikar@2wire.com

Abstract: Conference Key Management (CKM) is one of the primary issues in Secure Dynamic Conferencing (SDC). In this paper, we propose a novel CKM scheme for SDC based on the secret sharing principle and the novel concept/introduction of randomised access polynomial. Our scheme is simple, efficient, scalable, practical, dynamic and outperforms existing CKM schemes in overall comparison. Furthermore, if *t* or less users collude, the new scheme is unconditionally secure and able to defend against their collusions. The storage (O(1) at user end), computation and communication efficiency of the new scheme makes it well suited for the networks with low power devices.

Keywords: network security; secure dynamic conferencing; SDC; conference key management; CKM; secure group communication; SGC; group key management; GKM; access polynomial.

Reference to this paper should be made as follows: Zou, X. and Karandikar, Y. (xxxx) 'A novel Conference Key Management solution for Secure Dynamic Conferencing', *Int. J. Security and Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Xukai Zou is an Assistant Professor at the Department of Computer and Information Sciences at Indiana University-Purdue University Indianapolis, USA. He completed his PhD Degree from University of Nebraska-Lincoln in 2000. His research focus is in applied cryptography, network security, and communication networks.

Yogesh Karandikar is a Software Engineer with 2Wire Inc. He completed his Master of Science in Computer Science from Indiana University, Purdue University Indianapolis, USA. His research interests include network and communication security, application security, cryptography.

1 Introduction

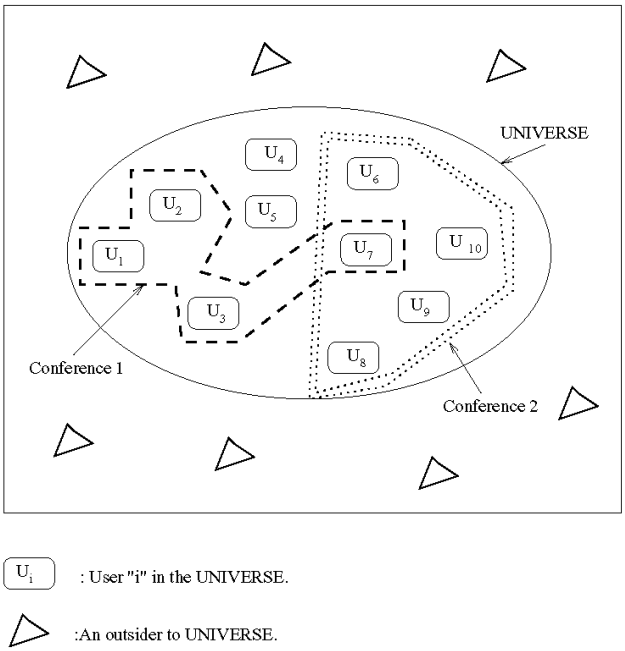
In real world, an event or a process where a group of people talk to each other is termed as a *conference*. Advances in technology allow a conference between people via telephone lines or over the internet. In the digital world, number of software programs or small sensor devices or grid elements interacting with one another can also be seen as examples of conferences. Thus we can define the term *conference* as an interaction between a group of entities that are part of the *universe*. We call the entities involved in a conference as *members* of the conference, entities in the universe but outside a conference as users and those out side the universe as outsiders throughout the paper. The universe can have many conferences going on simultaneously and a user can be a member of none, one or many conferences (see Figure 1).

In critical applications, a conference may require confidentiality, i.e., communication is limited to only those who are in the conference, but anyone else outside the conference cannot understand the communication even if he or she can intercept the communication. The same concept applies to conferences in the digital world. A conference must be secure such that only the members should be able to get the information shared. Encryption of data allows it to be transmitted in such a way that all can see the transmission but only a few, who have the decryption key, can understand it.

Also, conferences in the digital world are dynamic. There are two types of dynamics: join and leave. Users of the universe join a conference to become members or some outsiders join the universe as well as a conference. Some members leave a conference to become users or some members leave all the conferences as well as the universe to

become outsiders. In case of joins, the conference key needs to be changed in order to prevent the new joining members from accessing past data (backward secrecy). Similarly, in case of leaves, the conference key needs to be changed in order to prevent the leaving members from accessing future data (forward secrecy). The scenario of conferences described so far is defined as *Secure Dynamic Conferencing* (*SDC*).

Figure 1 An example of universe of users and conferences



High dynamics in SDC means that conference keys need to frequently be changed and distributed to the members. How to distribute keys to the members of conferences in a dynamic yet efficient manner is the biggest problem (we call it the problem of Conference Key Management (CKM)). In this paper we propose a simple, practical, scalable and efficient CKM scheme, based on polynomials over finite field. In Section 2 we discuss various schemes found in literature. Followed is our scheme in Section 3. We discuss security analysis and efficiency of our scheme comparing with other schemes in Section 4, followed by conclusion and future work.

2 State of the art

To our knowledge, just a few key distribution schemes for SDC have been proposed in literature. These schemes can be classified as a naive solution (Desmedt and Viswanathan, 1998), Public Key based SDC (PKSDC) schemes including the Chinese Remainder Theorem based SDC (secure lock) (SLSDC) scheme (Chiou and Chen, 1989), Symmetric Polynomial Based Schemes (SPSDC) (Blundo et al., 1998, 1993; Zou et al., 2002), the Interval based SDC scheme (ISDC) (Gouda et al., 2002), and the Key Tree based SDC scheme (KTSDC) (Adusumilli and Zou, 2005; Zou et al., 2004a, 2004b). Secure Group Communication (SGC) is a special case of SDC and it has been studied by many researchers (Beimel and Chor, 1994, 1996; Blom, 1985; Blundo and Cresti, 1995; Blundo et al., 1993, 1998; Burmester and Desmedt, 1995; Noubir, 1999; Noubir et al., 2002; Stinson, 1997). The book (Zou et al., 2004b) surveys many SGC schemes, but all the schemes for SGC can not be used for SDC directly.

The naive scheme proposed in Desmedt and Viswanathan (1998) assigns one (independent) key for each of $2^n - n - 1$ possible conferences (here n is the size of the universe), and gives each member $2^{n-1} - 1$ keys, one for each of the conferences the member can join. Whenever a member wants to communicate to the members in a conference, the member just picks up the key corresponding to the conference and does it. The main problems with the naive solution are its exponential number of keys to be stored with every user and no support for dynamics.

In PKSDC, whenever a member m_i wants to send a message M to a conference $C = \{m_{i_1}, \ldots m_{i_l}\} \cup \{m_i\}$, m_i selects a random session key k, encrypts the message with k, and encrypts k with these members' public keys $P_{i_1}, \ldots, P_{i_l}$ respectively, and broadcasts $(\{E_{P_{i_1}}(k), \ldots, E_{P_{i_l}}(k)\}, \{M\}_k)$ to the group. In SLSDC, the multiple encryptions of k, i.e., $\{E_{P_{i_1}}(k), \ldots, E_{P_{i_l}}(k)\}$ are combined into one value (called secure lock) using the Chinese Remainder Theorem. The SLSDC scheme has two advantages over PKSDC: a receiver can compute the key directly and efficiently from the lock value and conference members are hidden so that non conference members or outsiders cannot know who are in the conference. The main problems with this kind of schemes are:

- public key encryption and decryption are computationally expensive
- the requirement of encrypting a session key with every conference member's public key may cause scalability problem
- it requires the existence of a PKI.

In SPSDC (Blundo et al., 1993, 1998; Zou et al., 2002), each member is initially given certain secret information (a share of a symmetric polynomial), from which (along with some public information) the member can compute any conference key the member can join later. The main problem with this scheme is the exponential size of secret information.

The KTSDC schemes (Adusumilli and Zou, 2005; Zou et al., 2004a, 2004b) assume presence of a key tree (either centralised or distributed Adusumilli and Zou (2005)) and then use that key tree to distribute conference keys. The problem with these schemes is the overhead associated with having the tree. Also synchronisation of the key tree at every user upon dynamic operations is a big problem, considering unreliable nature of communication networks.

ISDC (Gouda et al., 2002) is similar to the above KTSDC scheme in the sense that it is also based on the key tree scheme. The main problem with the ISDC scheme is that the Group Controller (GC) relays all the conference messages, i.e., decrypting messages, encrypting messages (with multiple keys covering the conference members), and resending messages. Using the GC to relay messages is very naive and certainly inefficient and non-scalable.

We can enumerate the properties of a good solution to the problem of key distribution of SDC as follows:

- A solution should take care of user dynamics in an efficient manner.
- The scheme should be scalable for a large number of users and conferences.
- The conference keys should be independent in order to have an unconditionally secure scheme.
- Amount of storage required for keys or key materials, with every member, should be minimal.
- In some applications, the membership information of any conference should be secret. No one outside the conference should know who are in conference.

When we examine all the schemes proposed so far, each one of them lacks some property or other. In this paper we propose a polynomial based CKM scheme for SDC which satisfies all the above properties.

3 Proposed scheme

Finite fields are well studied in mathematics. A finite field over a prime q is denoted by F_q (Lausch and Nobaur, 1973; Lidl and Niederreiter, 1986). A typical univariate polynomial of degree t is represented as $a_0 + a_1 \times x + a_2 \times x^2 + \ldots + a_t \times x^t$. The coefficients $a_0, a_1, \ldots$ of a polynomial over a finite field F_q satisfy the property of $a_i < q$. In this section we describe our novel secret sharing idea based on polynomials over finite fields, by which we design and present our novel CKM scheme.

3.1 Secret sharing using polynomials over F_q

Let us assume that we want to share a secret K to m members over the internet in an efficient manner. We assume the presence of a GC who generates and distributes K. Like all existing centralised schemes, the new scheme assumes that there is a secure channel between the GC and each member. It should be noted that the secure channel is used only for the initial setup process and key management operations are performed by broadcasting/ multicasting (masked) key materials over an insecure channel.

The GC chooses a large prime q that forms the finite field F_q. It is assumed that $K \in F_q$. The GC selects a random polynomial $H(x)$ of degree t over F_q. Now, the GC computes another polynomial $S(x) = K - H(x)$. We call $H(x)$ and $S(x)$ as key polynomials. Let us consider a small example with $t = 4$. Let $H(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$, so the other key polynomial turns out to be $S(x) = (K - a_0) - a_1x - a_2x^2 - a_3x^3 - a_4x^4$. So evaluating key polynomials at $x = r \in F_q$ yields, $H(r) = a_0 + a_1r + a_2r^2 + a_3r^3 + a_4r^4$ and $S(r) = (K - a_0) - a_1r - a_2r^2 - a_3r^3 - a_4r^4$. Note that the coefficients of $S(x)$ are positive after modular arithmetic. Here they are shown as negative just for simplicity of illustration. More importantly, $H(r)$ and $S(r)$ are going to be values such that $H(r) + S(r) = K$, due to the unique construction of key polynomials. This is true for any $x = r \in F_q$, but it should be noted that $H(r_1) + S(r_2) \neq K$ if $r_1 \neq r_2$.

$S(x)$ and $H(x)$ need to be kept secret. In order for a valid user to compute $S(x = a)$ and $H(x = a)$ at some point $x = a$ so that the user can obtain K, another polynomial $h(x)$, called a masking polynomial, is introduced. The GC picks up a random t degree polynomial over F_q as $h(x)$ and keeps $h(x)$ secret too. The GC chooses a unique $ID_i \in F_q$ and computes $h(x = ID_i)$ for each valid user U_i. This $(ID_i, h(ID_i))$ is given to user U_i over the secure channel between U_i and the GC and it is called as U_i's secret. Basically each valid user gets a unique ID and a share of the masking polynomial. Let us continue with the example and assume the masking polynomial as $h(x) = a_5 + a_6x + a_7x^2 + a_8x^3 + a_9x^4$.

Now, the GC computes two public polynomials $W(x) = H(x) + h(x)$ and $P(x) = S(x) + h(x)$ which are broadcasted/multicasted over an insecure channel. From the coefficients of the public polynomials $W(x)$ and $P(x)$, it is impossible to guess the secret polynomials. Adding masking polynomials $h(x)$ makes sure that only those who have a share of $h(x)$ can get a share of the key polynomials from the public polynomials and no one else. This means that anyone (including outsiders) can see the public polynomials, but only valid users can get a share of the key polynomials. For example, evaluation of $W(x)$ at $x = r$ gives a value equal to $H(r) + h(r)$. Now to get $H(r)$ from $W(r)$ one has to know $h(r)$. Also note that a valid user can get only his or her share of the public polynomials. For example U_i can get $H(ID_i) = W(ID_i) - h(ID_i)$, however he or she can not get $H(ID_j)$ since $h(ID_j)$ is U_j's secret and no one else knows it. Similarly, U_i can (and only can) get $S(ID_i)$ from $P(x)$.

Thus the masking polynomial makes sure that only valid users can get shares of key polynomials. Also it makes sure that a valid user gets only his or her share of key polynomials. Thus only valid users can get the key from the broadcast/multicast messages. But conference key management for SDC requires the conference key to be shared with only members of a conference and not all valid users. In the next subsection we will propose an efficient CKM scheme for SDC by introducing a novel concept of randomised access polynomials.

3.2 Conference Key Management (CKM) for SDC

Similar to the above discussion, it is assumed that q is a large prime and t is a security parameter which determines collusion level. The GC selects a random t-degree masking polynomial $h(x)$ and also a random t-degree key polynomial $H(x)$. Let us further assume that there are n valid users in the

universe and each has received a unique secret (ID_i, $h(ID_i)$) from the GC. Suppose there are m valid users who want to have a conference. It is possible that many conferences are occurring simultaneously in the universe. The GC takes care of key management for all of them. For simplicity and to make the solution generic, we specify a conference by j and the conference key by K_j.

The GC computes the other key polynomial for conference j as $S_j(x) = K_j - H(x)$. It should be noted that one key polynomial ($H(x)$) remains constant and it is considered as a system polynomial. The GC distributes the secret $H(x)$ by broadcasting/multicasting $W(x) = H(x) + h(x)$. As explained earlier, all valid users can get a share of $H(x)$ from $W(x)$. Hence the other portion of key K_j needs to be published in such a way that only m members of the conference can get it and hence the key K_j. For doing that, the GC computes a randomised access polynomial for conference j using IDs of m conference members along $A_j(x) = (x - VID_j)\prod_{i=1}^{m}(x - ID_i) + 1$ with a virtual ID (a random integer).

Now, the GC publishes $P_j(x) = S_j(x) \times A_j(x) + h(x)$. This construction makes sure that only the members of a conference can get a share of $S_j(x)$ and hence the key. Any outsider or any other user outside the conference can not get the conference key. This is achieved by the construction of $A_j(x)$ which makes sure that $A_j(x = ID_r) = 1$ if and only if ID_r is a secret ID of a conference member or $x = VID_j$, otherwise $A_j(x = ID_r)$ = a random number. Thus a user U_r can get share of $S_j(x)$ as $S_j(ID_r) = P_j(ID_r) - h(ID_r)$ if he or she is the member of the conference (i.e., his or her ID is contained in the formation of $A_j(x)$), otherwise this operation returns just a random number. We will elaborate the security of the new scheme in Section 4. It is worthy to note that the virtual term ($x - VID$) is changing every time $A(x)$ is computed, thus, $A_j(x)$ will be different even though the members in two conferences are the same.

3.2.1 Key derivation

Any valid user U_i can get the value $H(ID_i) = W(ID_i) - h(ID_i)$. To get conference key K_j, U_i needs to get $S_j(ID_i)$. If U_i is a member of conference j, he or she can get $S_j(ID_i) = P_j(ID_i) - h(ID_i)$ and then get the key as $K_j = H(ID_i) + S_j(ID_i)$. The randomised access polynomial makes sure that no user outside the conference can get share of $S_j(x)$.

3.2.2 Dynamics

There are two possible cases of dynamics, user joins and member leaves. The scheme can deal with both of them in an efficient manner.

- *The join operation*. There are two scenarios to consider.

  First, when a user joins a conference j, the GC changes the conference key K_j to K_j' to maintain backward secrecy. The GC computes the new key polynomial as $S_j'(x) = K_j' - H(x)$. Note that the system key polynomial $H(x)$ does not change. The GC constructs a new randomised access polynomial $A_j'(x)$ by including the joining user's ID and using a new VID_j' in the computation. Then the GC publishes $P_j'(x) = S_j'(x) \times A_j'(x) + h(x)$. Note that the new member to a conference can not get the previous conference keys since his or her ID was not used to compute previous randomised access polynomials. He or she can get current key as he or she was included in the conference and future keys as long as he or she remains in the conference.

  Second, when an outsider joins universe and also a conference j, the GC finds a random unique ID_{new} and calculates $h(ID_{new})$. The new user gets his or her secret as (ID_{new}, $h(ID_{new})$) from the GC. Once he or she becomes a valid user in universe, he or she can join a conference in the same way as the operation discussed above.

- *The leave operation*. We consider two scenarios here too.

  First, when a member leaves a conference j, the GC changes key K_j to K_j' to maintain forward secrecy. The GC computes the new key polynomial as $S_j'(x) = K_j' = H(x)$. Also the GC computes the new randomised access polynomial $A_j'(x)$ by excluding the ID of the leaving member and replacing VID_j with a new VID_j' and broadcasts $P_j'(x) = S_j'(x) \times A_j'(x) + h(x)$. Thus the leaving member can not get the conference data anymore.

  Second, when a member leaves the entire universe and becomes an outsider, the GC takes the member out of the conference (if the member is in) same as the above leave operation and removes his or her name from the universe list.

It is possible that multiple users join and/or multiple members leave simultaneously. This can be efficiently conducted as for the single join and/or leave: just including the new joining members' IDs in and excluding the leaving members' IDs out of the formation of the new randomised access polynomial (again, a new VID' is used).

Thus our scheme handles user dynamics in an elegant and efficient manner.

4 Discussion

In this section we provide security proofs of our scheme. We also discuss the efficiency of our scheme in terms of storage, computation, and communication complexities. The comparison of our scheme with existing schemes is also presented based on the desirable properties of a good solution for SDC CKM and efficiency.

4.1 Security analysis

Like all other secret sharing-based schemes, the new scheme also assumes that t is the maximum number of possible users who attempt to collude. Let us assume that the GC has setup the system by choosing a secret masking polynomial $h(x)$ and a secret key polynomial $H(x)$ and publishes $W(x) = H(x) + h(x)$ (they are all of degree t). The GC chooses a unique ID_i and computes $h(ID_i)$ for each user U_i, in the universe. Every U_i gets $(ID_i, h(ID_i))$ as his or her secret. It is assumed that $h(x)$ (similarly, $H(x)$) is only known to the GC. From Shamir's (1979) secret sharing principle, t or less users collude with their $(ID_i, h(ID_i))$ but they cannot gain any bit of information about $h(x)$. That is to say, $h(x)$ is unconditionally secure if t or less users collude, so does the new scheme. On the other hand, $t + 1$ or more users collude and they can recover entire $h(x)$ by using polynomial interpolation over $t + 1$ points $\{(ID_i, h(ID_i))\}$. In summary, we could feasibly assume:

- t is selected according to application properties and security requirement such that the maximum number of colluding users will not exceed t
- due to the efficiency of the new scheme, t can be selected to be large to prevent colluding attacks without affecting efficiency
- in case more than t users are found to attempt colluding, the GC can discard the current $h(x)$ and generate a new $h(x)$ and distribute the new shares to legitimate users.

Furthermore, periodically refreshing $h(x)$ can be embedded in the new scheme to prevent collusion attempts.

As for attacks, the possible information which can be exploited is $W(x)$ and $P(x)$. As for $W(x)$ $(= H(x) + h(x))$ which is unchanged once published, attackers cannot get $H(x)$ or $h(x)$ without knowing the other. The possible values the users can get are nothing but the shares $H(ID_i)$ from $h(ID_i)$. Thus, $W(x)$ does not help for attacking. As for $P(x)$, none of $A(x)$ and $S(x)$ can be obtained from ONE $P(x)$ because $h(x)$ is contained in $P(x)$. Suppose two or more $P(x)$ such as $P_j(x)$ and $P_{j+1}(x)$ are observed. $h(x)$ can be cancelled out by differentiating $P_{j+1}(x)$ and $P_j(x)$ as $P_{j+1}(x) - P_j(x) = S_{j+1}(x)A_{j+1}(x) - S_j(x)A_j(x)$. However, $S_{j+1}(x)$ and $S_j(x)$ are random polynomials, thus, being independent and $A_{j+1}(x)$ and $A_j(x)$ are not equal (Note: $A_{j+1}(x)$ contains a virtual term $(x - VID_{j+1})$ and $A_j(x)$ contains a virtual term $(x - VID_j)$, so they will still be different even when they contains same user IDs). Thus, a malicious user cannot figure out any of $A_j(x)$, $S_j(x)$, $A_{j+1}(x)$, and $S_{j+1}(x)$. Furthermore, even though multiple (less than or equal to t) users collude, their collusion will not generate any further information. This is because all polynomials $A_j(x)$, $S_j(x)$, $A_{j+1}(x)$, and $S_{j+1}(x)$ will generate useful information only when x in all of them is bound to a same value, but the IDs of colluding users are different. As a result, $P(x)$s do not help for attacks either. As for both $W(x)$ and $P(x)$, due to the same reason of same x-value binding, their combination will not help for attack. In summary, the new scheme is secure against the attacks from malicious users (except the collusion of more than t users). It is obvious that all attacks from outsiders, regardless of a single attacker or multiple attackers, will not success either.

Next we discuss the correctness and security of our scheme via an example shown in Figure 1. It shows some users in the Universe, some outsiders (triangles in the Figure), and two conferences. Conference 1 consists of four members U_1, U_2, U_3 and U_7 and conference 2 consists of members U_6, U_7, U_8, U_9 and U_{10}. Let K_1 be the key of conference 1. The GC computes key polynomial $S_1(x) = K_1 - H(x)$. The GC computes the randomised access polynomial $A_1(x)$ using IDs of members in conference 1 plus a random virtual ID, i.e., ID_1, ID_2, ID_3, ID_7 plus VID_1 and publishes $P_1(x) = S_1(x)A_1(x) + h(x)$.

Let us see what happens if a user U_4 who is not in conference 1 tries to get K_1. U_4 can get $H(ID_4) = W(ID_4) - h(ID_4)$ using public information and his or her secrets. Now he or she needs $S_1(ID_4)$ to get key K_1. If U_4 evaluates $P_1(x)$ at $x = ID_4$, he or she gets $P_1(ID_4) = S_1(ID_4) \times A_1(ID_4) + h(ID_4)$. From this he or she can get value of $S_1(ID_4) \times A_1(ID_4)$ by subtracting his or her secret $h(ID_4)$ from $P_1(ID_4)$. Still he or she can not get the value of $S_1(ID_4)$ because the randomised access function is secret. Also the list of members in the conference is secret so no one can construct the randomised access polynomial either. For any member of the conference, the randomised access function evaluates to 1 and they get the share of $S_1(x)$ easily. If an outsider tries to get the key, the masking polynomial prevents him/her from getting any key share. Thus our scheme is secure against attacks from outsiders as well as users in universe that are not members of a conference.

Let us see what happens if someone such as U_4 tries to guess different values of x to get the key. Attacker U_4 can guess $x = r$ for which randomised access polynomial $A_1(x = r) = 1$. But then he or she gets $P_1(r) = S_1(r) + h(r)$. From this he or she can not get $S_1(r)$ without knowing $h(r)$. The masking polynomial $h(x)$ is secret with the GC and any user knows only his or her share of it, thus it is impossible for the attacker to get share of $S_1(x)$. Similarly, for an outside attacker, the guess attack yields nothing. Thus our scheme is secure against guess attacks as well.

4.2 Performance analysis

Our scheme is efficient in terms of storage, computation and communication efficiency. Each user U_i's secret consists of $(ID_i, h(ID_i))$, which is quite small and requires $O(1)$ space. The GC has to store all users IDs, secret masking polynomial and the key polynomial as well as information about conferences. Generally the GC is assumed to have a lot of computing power hence this storage is not much for the GC.

The conference key computation for a member of a conference requires two polynomial evaluations at $x = ID_i$, two subtractions, and one addition. Since $P(x)$ to be

evaluated is of degree $t + m + 1$ (note: 1 here is the contribution of the virtual term $(x - VID)$), the key computation complexity is $O(t + m)$.

The GC broadcasts the key materials in terms of two polynomials: one of $t + m + 1$ degree and other of degree t. To broadcast a polynomial, it is required to broadcast its coefficients. Thus the communication complexity is $O(t + m)$ which is small as well.

Thus it can be seen that our scheme is very efficient. Efficiency of our scheme makes it suitable for using even in wireless or sensor networks which involve devices with moderate computing resources.

4.3 Comparison

Here we compare our proposed scheme with other key management schemes for SDC. Table 1 shows this comparison in terms of properties of a good solution discussed in Section 2. It is important for a good scheme to have all the properties, but all existing schemes lack one property or the other. Our proposed scheme is the only scheme which meets all the properties and hence is qualitatively better than any other existing scheme.

Table 1 Comparison of typical SDC schemes

Scheme	*Dynamic*	*Scalable*	*CCK**	*Secrecy***
Naive	No	No	No	YES
PKSDC	Yes	No	Yes	No
SLSDC	Yes	No	Yes	Yes
SPSDC	Yes	No	No	YES
ISDC	Yes	No	Yes	NO
KTSDC	Yes	Yes	Yes	No
KTDCKM-SDC	Yes	Yes	Yes	No
Our Scheme	Yes	Yes	Yes	Yes

*Changeable conference keys.

**Secrecy about conferences and members of conferences.

Our scheme is better than existing schemes in a quantitative comparison as well. Table 2 shows the comparison of storage (at user end), key computation (at user end) and communication (between the GC or conference initiator and members) complexities. Storage complexity is calculated in terms of storage required at user end. Similarly computation complexity is calculated in terms of computations required by users for getting key. Communication complexity is based on size of broadcast messages. It can be easily seen from the complexity comparison table that our scheme is independent from the size n of the universe and lot more efficient than any other existing schemes. The efficiency of our scheme makes it portable to wireless ad-hoc or sensor networks, which is the future direction of research.

Table 2 Complexity comparison

Scheme	*Storage*	*Computation*	*Communication*
Naive	$O(2^{n-1} - 1)$	$O(1)$	$O(2^{n-1} - 1)$
PKSDC	$O(n)$	$O(mP)$*	$O(n)$
SLSDC	$O(n)$	$O(mP)$*	$O(m)$*
SPSDC	$O(n^m)$	$O(t^2)$	$O(n^m)$
ISDC	$O(\log(n))$	$O(D\log(n))^+$	$O(\log(n))$
KTSDC	$O(\log(n))$	$O(D\log(n))^+$	$O(\log(n))$
KTDCKM-SDC	$O(n)$	$O(D\log(n))^+$	$O(\log(n))$
Our Scheme	$O(1)$	$O(t + m)$	$O(t + m)$

^{n}Total number of users in Universe.

^{m}Number of members in a typical conference.

^{t}Degree of polynomials used.

*P is the complexity of public key encryption/decryption.

+D is the complexity of secret key encryption/decryption.

5 Conclusion

We proposed a novel conference key management scheme for secure dynamic conferencing, based on polynomials over finite field. Our scheme is simple, practical, scalable, and efficient in terms of storage, computation and communication. Our scheme takes care of user dynamics in an elegant and efficient way and is able to hide conference membership. Thus our scheme solves all the problems that existing SDC CKM schemes have. Efficiency of our scheme in terms of storage and key computations at user end makes it suitable to use in networks with low power devices. Future work includes performance evaluation and experimentation to judge applicability of the scheme in wireless and sensor networks.

Acknowledgement

The authors sincerely appreciate the constructive comments from journal editors and anonymous reviewers. This work was partially supported by the US NSF grant CCR-0311577.

References

Adusumilli, P. and Zou, X. (2005) 'KTDCKM-SDC: a distributed conference key management scheme for secure dynamic conferencing', *Proceedings of the Tenth IEEE Symposium ON Computers and Communications (ISCC)*, Cartagena, Spain, pp.476–481.

Beimel, A. and Chor, B. (1994) 'Interaction in key distribution schemes', *Advances in Cryptology – CRYPTO'93, LNCS*, Springer, Berlin, Vol. 773, pp.444–457.

Beimel, A. and Chor, B. (1996) 'Communications in key distribution schemes', *IEEE Transactions on Information Theory*, Vol. 42, pp.19–28.

Blom, R. (1985) 'An optimal class of symmetric key generation systems', *Advances in Cryptology – EUROCRYPT'84, LNCS*, Springer, Berlin, Vol. 209, pp.335–338.

Blundo, C. and Cresti, A. (1995) 'Space requirements for broadcast encryption', *Advances in Cryptology – EUROCRYPT'94, LNCS*, Springer, Berlin, Vol. 950, pp.287–298.

Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U. and Yung, M. (1993) 'Perfect secure key distribution for dynamic conferences', *Advances in Cryptology – CRYPTO'92, LNCS*, Springer, Berlin, Vol. 740, pp.471–486.

Blundo, C., Mattos, L.A.F. and Stinson, D.R. (1998) 'Generalized Beimel-Chor scheme for broadcast encryption and interactive key distribution', *Theoretical Computer Science*, Vol. 200, Nos. 1–2, pp.313–334.

Burmester, M. and Desmedt, Y. (1995) 'A secure and efficient conference key distribution system', *Advances in Cryptology – EUROCRYPT'94, LNCS*, Springer, Berlin, Vol. 950, pp.275–286.

Chiou, G.H. and Chen, W.T. (1989) 'Secure broadcasting using the secure lock', *IEEE Transactions on Software Engineering*, Vol. 15, No. 8, pp.929–934.

Desmedt, Y. and Viswanathan, V. (1998) 'Unconditionally secure dynamic conference key distribution', *Proceedings of the IEEE International Symposium on Information Theory*, Cambridge, MA, USA, pp.383–383.

Gouda, M.G., Huang, C-T. and Elnozahy, E.N. (2002) 'Key trees and the security of interval multicast', *Proceedings 22nd International Conference on Distributed Computing Systems*, pp.467, 468.

Lausch, H. and Nobaur, W. (1973) *Algebra of Polynomials*, North Holland Publishing Company, North Holland, Amsterdam.

Lidl, R. and Niederreiter, H. (1986) *Introduction to Finite Fields and their Applications*, Cambridge University Press, New York, NY, USA.

Noubir, G. (1999) 'A scalable key distribution scheme for dynamic multicast groups', *The Third European Research Seminar on Advances in Distributed Systems*, Available at http://www.ccs.neu.edu/home/noubir/publications/N99.pdf.

Noubir, G., Zhu, F. and Chan, A.H. (2002) 'Key management for simultaneous join/leave in secure multicast', *IEEE International Symposium on Information Theory (ISIT)*, pp.325–330.

Shamir, A. (1979) 'How to share a secret', *Communication of ACM*, Vol. 22, pp.612–613.

Stinson, D.R. (1997) 'On some methods for unconditionally secure key distribution and broadcast encryption', *Design, Codes and Cryptography*, Vol. 12, pp.215–243.

Zou, X., Magliveras, S. and Ramamurthy, B. (2002) 'A dynamic conference scheme extension with efficient burst operation', *Congressus Numerantium*, Vol. 158, pp.83–92.

Zou, X., Magliveras, S. and Ramamurthy, B. (2004a) 'Key tree based scalable secure dynamic conferencing schemes', *Proceedings of International Conference on Parallel and Distributed Computing and Systems (PDCS 2004)*, MIT Cambridge, MA, USA, 9–11 November, pp.61–66.

Zou, X., Ramamurthy, B. and Magliveras, S.S. (Eds.) (2004b) *Secure Group Communications over Data Networks*, Springer, ISBN: 0-387-22970-1 (The ebook ISBN: 0-387-22971-X).