SBIBD 네트워크에서 다자간 원격회의를 위한 회의용 키 생성 프로토콜

김성열* · 김동현**

Conference Key Agrement Protocol for Multilateral Remote Conference Employing a SBIBD Network

Seong-yeol Kim* · Dong-Hyun Kim**

요 약

다자간 회의용키 협의는 제 3의 신뢰기관 없이 다수의 구성원 참여에 의해 회의용 세션키를 생성함으로 말한다. 이 논문에서는 SBIBD(symmetric balanced incomplete block design)를 이용한 효율적인 회의용 키협의 프로토콜을 제안한다. 제안된 방식은 회의 참여자간 교환하는 메시지의 양과 메시지교환 횟수를 최소화하고 각 참여자가 균등한 메시지 오버헤드를 갖도록 한다.

제안된 프로토콜에서는 v명의 참여자가 있고 각 참여자 i가 생성한 랜덤넘버가 R_i라고 할 때 Diffe-Helman의 세션키 생성법을 변형한 $\Pi_{i=0}^{v-1}R_i$ 형태의 키를 생성하게 된다. 이때 메시지 교환횟수는 3회, 메시지오버헤드는 $O(v\sqrt{v})$이 된다. 또한 이 프로토콜은 이산대수문제에 근거하여 안전성을 보장한다.

ABSTRACT

A conference key agreement system is a scheme to generate a session key in a contributory manner in order to communicate with each other securely among participants. In this paper an efficient conference key agreement system is proposed by employing symmetric balanced incomplete block design(SBIBD), one class of block designs. The protocol presented not only minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key.

Our protocol constructs a conference key which takes modified Diffe-Helman form of $\Pi_{i=0}^{v-1}R_i$, where v is the number of participants and R_i is a random number generated from member i. In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is $O(v\sqrt{v})$. Our protocol can be proved as computationally difficult to calculate as discrete logarithms.

키워드

회의용 키, SBIBD, 키 생성 프로토콜, 다자간 회의

Ⅰ. Introduction

With the emergence of many group-oriented distributed applications, such as from tele/video

* 울산과학대학 컴퓨터정보학부

** 순천청암대 부동산학과

접수일자 : 2009. 10. 19

심사완료일자 : 2009. 11. 20

conference to usn there is a need for security services to provide group-oriented communication privacy and data integrity. To provide this form of group communication privacy, it is important that group members can establish a common secret key for encrypting group communication data[1]-[3].

Conference key can be generated by two different types.

One is centralized method. In this case, TTP(Trusted Third Party) or single member generates a key and distributes it. Simple but this has significant drawbacks such as overall reliance on a single party. The other is contributory method. Each group member contributes to generate a key independently. This is a process of key agreement among members. Conference key agreement systems can be classified into two categories according to the group feature whether it is for static group or dynamic group. Conference key agreement for dynamic group needs to consider group modification such as member addition as well as initial key agreement. In case that conference key agreement is performed on complete graph, it requires $v(v-1)$ messages to be sent and one round message exchange, where v is the number of participants[4]. It is tried to construct efficient key agreement protocols by extending Diffie-Hellman[5] to groups in many researches. [4] is the first attempt to provide contributory key agreement. Here, they proposed a method peformed on a logical ring network. It requires $v-1$ rounds message exchange, v^2 of traffic overhead and $v^2/2$ of exponentiation, Another remarkable result is [6] in which it takes only 2 round message exchange, but it requires $2v$ times broadcasting and n^2 of exponentiation. [6] looks very efficient but it requires each entity to receive $v-1$ messages in a single round. [7] is another contributory key agreement system which requires $2v$ rounds message exchange,

$2v$ times unicasting and 2 times broadcasting of traffic overhead and $4v$ of exponentiation, but each entity needs still to receive $v-1$ messages in each broadcasting round.

In this paper an efficient conference key agreement system for a static group is presented by employing symmetric balanced incomplete block design(SBIBD), one class of block designs. The protocol presented not only minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key. Our protocol constructs a conference key which takes the forms of $\Pi_{i=0}^{v-1} R_i$, where v is the number of participants and R_i is a random number generated from member i. In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is $O(v\sqrt{v})$. Our protocol can be proved as computationally difficult to calculate as discrete logarithms.

The rest of this paper is organized as follows. Section 2 defines SBIBD. Section 4 presents our conference key agreement protocol which is performed on a $(v,\ k+1,\ 1)$-configuration and analyse the security and complexity issues. This paper concludes with summary and on-going and future work in section 4.

Ⅱ. Definition of (v,k,λ)-configuration and its generation

Let $V=\{0,1,...,\mathrm{v}-1\}$ be a set of v elements. *Let* $B=\{\mathrm{B}_0,\mathrm{B}_1,...,\mathrm{B}_{\mathrm{b}-1}\}$ be a set of b blocks, where B_i is a subset of v and $(|\mathrm{B}_{\mathrm{i}}|=\mathrm{k})$. For a finite incidence structure $\sigma=\{\mathrm{V},\mathrm{B}\}$, if σ satisfies following conditions, then it is a balanced incomplete block design[8][9], which is called a (b,v,r,k,λ)-configuration.

1. b is a collection of b k-subsets of v and this k-subsets are called the blocks.

2. Each element of v is related with exactly r of b blocks.
3. Every two objects of v appears simultaneously in exactly λ of b blocks.
4. $k < v$.

For a (b,v,r,k,λ)-configuration, if it satisfies $k = r$ and $b = v$, then it is a symmetric balanced incomplete block design (SBIBD)[8][9] and it is called a (v,k,λ)-configuration.

As shown above, it is not true that there exists a BIBD or SBIBD for arbitrary set of parame- ters b,v,r,k and λ. However there is no known sufficient condition on the existence of a certain (b,v,r,k,λ)-configuration or (b,v,r,k,λ)-configuration.

Our key agreement system is based on the feature of $(v,k+1,1)$-configuration, that is, each block has $k+1$ elements and every two object appears simultaneously only one time in v blocks. In the research[10], they proposed an efficient method for generating an incidence structure $\sigma = \{\mathrm{V,B}\}$ satisfying the condition for a $(v,k+1,1)$-configuration in the case that k is a prime number and proved it. We can make use of it to devise an efficient and fair key agreement protocol.

III. Design of a Conference Key Agreement System on $(v,k+1,1)$-configuration

An efficient conference key agreement system is now constructed on $(v,k+1,1)$-configuration generated from Algorithm of [10]. In our protocol, every member contributes evenly to compute the same conference key, $K = \Pi_{n=0}^{v-1} R_n$, which is computationally difficult to calculate as discrete logarithms, by 3 rounds message exchange and $v\sqrt{v}$ traffic overhead, where v is the number of participants and R_n is a random number generated from member n. Algorithm 1 is the conference key agreement system we propose. Notation used in this paper is as Table 1.

Table 1. Notation

notation	meaning
X_i	i^{th} set in a family of set X
$X_{i,j}$	j^{th} element of X_i
X^i	i^{th} member of family of sets X in which each member is a family of sets j^{th} set in a family of sets X^i
X^i	j^{th} set in a family of sets X^i
$\sigma\{\mathrm{V,X}$	an incidence structure where V is a set, X is a family of sets and X_i is a subset of V

Algorithm 1:
Construction of a Conference Key Agreement

input: N : a prime number
g : a primitive element

$g \in Z_N$
$Z_N = \{0, g, g^2, ..., g^{N-1} = 1\}(\mathrm{mod N})$
V : the set of participants
X={V,C} : {v,k+1,1}-configuration generated from Algorithm 2

output: each member computes the same key K.

1. Each member n on V defines two sets $S1_n$ and $S2_n$ as below.
2. Each member n generates a random number R_n and $Q_n = g^{R_n}(\mathrm{mod N})$.
3. Each member n sends Q_n to each member on $S2_n$
4. Each member n generates $M_i = \backslash g^{R_n}$, $R_n \times g^{R_n R_i}\}(\mathrm{mod N})$
5. Each member n computes $K = R_n \prod R_i(\mathrm{mod N})$, where $i \in S2_n$.

6. Each member n computes $pK_{n,i} = g^{R_n R_i} \prod R_j (\mathrm{mod} N)$, where $i \in S2_n$, $j \in \{\mathrm{C_n} - \{\mathrm{i}\}\}$ and sends $pK_{n,i}$ to member i on $S2_n$.

7.Each member computes the same conference key $K = K \prod pK_{i,n} (\mathrm{mod N})$, where $i \in S1_n$.

theorem 1. According to Algorithm 1, every member of V computes the same conference key $K = \prod R_n$, where $n \in V$.

proof. Let an arbitrary member n define $S1_n = \mathrm{a_{1,}a_{2,}...a_k}$ and $S2_n = \mathrm{b_{1,}b_{2,}...b_k}$. Cardinality of the set $S1_n$ and $S2_n$ is k because every block contains $(k+1)$ elements and every element appears on $(k+1)$ blocks in a $(v,k+1,1)$ -configuration.

For an element a on $S1_n$, it is true that if a is a member of $S1_n$ then n is a member of $S2_a$. Similarly, for an element b on $S2_n$, if b is a member of $S2_n$ then n is a member of $S1_b$. This means member n receives messages from members on $S1_n$ while it sends messages to members on $S2_n$ and receives from members on $S2_n$ while sending to members on $S2_n$

Table 2. Computation of the conference key on member 7 on X = {V, C}

round	receiving messages or computation

define	S1$_7$ = {1, 8, 12}
	S2$_7$ = {2, 6, 11}

1$^{\mathrm{st}}$	g^{R1}, g^{S}, g^{R12},
2$^{\mathrm{nd}}$	$\{\mathrm{g^{R2}, R_2 \times g^{R2R7}}\}$, $\{\mathrm{g^{R6}, R_6 \times g^{R6R7}}\}$, $\{\mathrm{g^{R11}, R_{11} \times g^{R11R7}}\}$
compute	$x = g^{R2^{R7}}; R_{2 = (R2} \times g^{R2R7})/x$ $x = g^{R6^{R7}}; R_6 = (R_6 \times g^{R6R7})/x$ $x = g^{R11^{R7}}; R_{11} = (R_{11} \times g^{R11R7})/x$ $K = R_7 \times R_2 \times R_6 \times R_{11}$
3$^{\mathrm{rd}}$	$pK_{1,7}, pK_{8,7}, pK_{12,7}$
compute	$K = K \times (pK_{1,7}/g^{R1R7}) \times (pK_{8,7}/g^{R8R7}) \times (pK_{12,7}/g^{R12R7})$ because $pK_{1,7} = g^{R1R7} \times R_1 \times R_4 \times R_{10}$, $pK_{8,7} = g^{R8R7} \times R_0 \times R_8 \times R_9$ and $pK_{12,7} = g^{R1.2R7} \times R_3 \times R_5 \times R_{12}$

Member n can compute $\prod R_b$, multiplication of $(k+1)$ random numbers, from the messages received in the 2^{nd} round, where $b \in C_n$. The message $pK_{a,n}$, arrived in the 3^{nd} round, is consists of k random numbers generated from members on $C_a - \mathrm{n}$, where $a \in S1_n$. So member n come to gain multiplication of k^2 random numbers and each of these k^2 random numbers has individual source because every two block has only one common element in a $(v,k+1,1)$-configuration and this element is n. Therefore arbitrary member n computes the same conference key $K = \prod R_n$, where $n \in V$.

theorem2. The key computed from Algorithm 3 is computationally difficult to calculate as discrete logarithms.

Proof. Three rounds message exchanging is performed in algorithm 1 for key agreement. In the first round, while a member n is sending g^{R_n} to member j, he/she receives g^{R_i} from member i. In the second round, member n receives g^{R_j} and $Y = R_j \times g^{R_n R_j}$. It is possible for the member n to obtain R_j from Y because of $R_j = \mathrm{Y}/(\mathrm{g^{R_j}})^{\mathrm{R_n}}$. But this critical information R_j can be protected because the thing provided to eavesdropper is only g^{R_j} and g^{R_n}. message M can be calculated by only member n in the same manner. Therefore finding the key generated from this algorithm is a discrete logarithm problem.

Ⅳ. Conclusion

An efficient key agreement system is presented for group communication. Our protocol minimizes the message overhead and message exchanging rounds but also makes every participant contribute evenly for generating a conference key and it is computationally difficult to calculate as discrete logarithms for an eavesdropper to find the key.

Proposed protocol constructs a conference key which takes the forms of $\Pi_{i=0}^{v-1} R_i$, where v is the number of participants and R_i is a random number generated from member i. In a special class of SBIBD, it takes only 3 rounds message exchange and message overhead is $O(v\sqrt{v})$, where every member sends and receives k messages in each round equally.

This algorithm is well performed when the number of participants is $v=k^2+k+1$. We are studying the method to apply our protocol in the case of arbitrary number of participants.

References

[1] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups", IEEE Transactions on Parallel and Distributed Systems, August 2000.

[2] Patrick P. C. Lee, John C. S. Lui and David K. Y. Yau, "Distributed Collaborative Key Agreement Protocols for Dynamic Peer Groups", IEEE International Conference on Network Protocols, 2002.

[3] M. Eltoweissy, M. Moharrum and R. Mukkamala, "Dynamic Key Management in Sensor Network", IEEE Communication Magazine, 2006

[4] I.Ingemarrson, D. T. Tang and C. K. Wong, "A conference key distribution System", IEEE Trans. Inform. Theory Vol.28, pp.714-720, 1982.

[5] Whit Diffie and Martin Hellman, "New Direction in cryptography", IEEE Trans Inform. Theory, Vol.22, No.6, pp.644-654, 1976.

[6] Burmester and Y. Desmedt, "A Secure and Eff icient Conference Key Distribution System", LNCS Vol.950, pp.275-286, 1994.

[7] M Steiner, G.Tsudik and M.Waidner, "Diffie- Hellman Key Distribution Extended to Groups", ACM CCS96, pp.31-37, 1996.

[8] M. K. Bennett, "Affine and projective geometry", Wiley & Sons, 1995.

[9] C. L. Liu, "Introduction to Combinatorial Mathamatics", McGraw-Hill, NY, pp.359-383, 1968.

[10] O. Lee, M. Anshel, and I. Chung, "Design of an efficient load balancing algorithm on distributed networks by employing symmetric balanced incomp[lete block design", IEEE Proc-Commun, Vol. 151, No.6, pp.535-538, 2004.

저자 소개

김성열(Seong-yeol Kim)

1994년 조선대학교 전자계산학과 (이학사)
1996년 조선대학교대학원 전자계산학과 (이학석사)
2000년 조선대학교대학원 전자계산학과 (이학박사)
2002년 ~ 현재 울산과학대학 컴퓨터정보학부 부교수
※관심분야 : 정보보안, 분산시스템, 전자상거래, 무선인터넷, 임베디드 시스템

김동현(Dong-hyn Kim)

1992년 광운대학교 공학석사
2002년 조선대학교 이학박사
2004년 행정학 석사
2008년 부동산학 박사 수료
1996년~현재 순천청암대학교 교수
※관심분야 : 컴퓨터응용, 디지털컨텐츠, 전자상거래, 부동산 정보