Diffie-Hellman Technique Extended to Efficient and Simpler Group Key Distribution Protocol

Vankamamidi S. Naresh
Department of Computer Science
S.V.K.P. and Dr. K.S.Raju.
Arts and Science College
Penugonda-534320,India

Nistala V.E.S. Murthy
Department of Computer Science
and Systems Engineering,
Andhra University
Visakhapatnam-530003,India

ABSTRACT
Ever since 2-party Diffie-Hellman exchange was first proposed in 1976, there have been efforts to extend its simplicity and elegance to a group setting. Notable solutions have been proposed by michael Steiner Gene Tsudik Waidner(in 1996) and Recently G.P.Biswas was proposed a contributory group key agreement protocol for generation of multiparty key and compared with other protocol and satisfactory results obtained.

In this paper an m-party DH key distribution for group (improved group DH) was proposed by modifying G.P.Biswas protocol and we argued that our protocol is optimal with respect to most of the aspects of protocol complexity and also it's security discussed.

Keywords
Diffie-Hellman technique, DDH problem, key distribution, key exchange operations, secure data transmission.

1. INTRODUCTION
It has been almost 33 years since Diffie - Hellman (DH) 2-party key exchange was first proposed in [1]. In the mean time, there have many attempts to extend its elegance and simplicity to the group setting. The main motivating factors are the increasing popularity of various types of group ware applications such as conference calls, distributed computation, distributed databases and so on in a secure way since the key distribution is the cornerstone of secure group communication, it has naturally received a lot of attention.(see for example:[2],[3],[4],[5],[6],[7], 8],[9],[10], [11]).

Recently Biswas proposed an efficient contributory multiparty key exchanging technique for a large static group is proposed. In this technique a member, who acts as a group controller forms two-party groups with other group members and generates a DH-style shared key per group. It then combine s these key into a single multi-party key and acts as a normal group member. This technique has been compared with other multiparty key generating techniques and satisfactory results have been obtained. But In this Group DH protocol the group controller needs to execute comparatively more KEOs than other group members, so the key generation may be delayed and also the over all delay of key generation increase with the increase in the number of parties.

To overcome above problem, In this paper we proposed a Improved group DH protocol for generation of m-party key by modifying Biswas Group DH protocol, with reduction of exponential operation.

The main advantages of the proposed work are:
(1). It uses simpler steps and only two steps.
(2). Needs comparatively less communication and computational cost.
(3) Reducing overall delay of key generation by reducing computational load on group controller

Generation of multi-party key:
Although the original DH generates a shared key in a two party –group, different researchers have extended it to the multi-party situations. In this section, we will discuss a useful DH-based multi-party key-generating technique for large static groups, called Group-DH. We consider that it holds DDH assumption for the security of the group key.

1.1 Review of the Group-DH technique
Group-DH technique was a contributory group key agreement protocol to exchange a multi-party key among the group members. In Group-DH, an arbitrary group member acts as a group controller that actually establishes the group-key and after which it becomes a normal member of the group. Let the group controller be P_i , where $1 \leq i \leq n$ for n-party group.
Initially it itself forms a two-party group with each of the remaining group members, and produces (n -1) two-party groups. The P_i then generates a DH style key per group, and produces (n-1) shared keys for (n -1) two-party groups. In order to accomplish it, P_i generates a public key and broadcasts to the remaining group members.

The public key X_i can be generated using

$$X_i = g^{x_i} \bmod n \qquad (1)$$

where x_i is the private key of the P_i.
Each group member, P_j, where $j \neq i$ also assumes a private key and generates a public key as

$$X_j = g^{x_j} \bmod n \qquad (2)$$

where x_j is the private key **of** P_j and $1 \leq j \leq n$, $j \neq i$.
Each P_j then transmits X_j to the group controller, P_i . After exchanging the public keys, each member similar to the basic DH generates a unique shared key, K_i with group controller as

$$K_i = X_i^{x_j} \bmod n = g^{x_i x_j} \bmod n \qquad (3)$$

Similarly, P_i generates the same shared key, using

$$K_i = X_j^{x_i} \bmod n = g^{x_j x_i} \bmod n \quad (4)$$

It actually generates (n-1) shared keys for (n-1) groups. The group controller combines these shared keys to produce a single group key. The P_i computes the public key X_k as given below and sends to P_j.

$$X_k = g^{\prod K_{k \neq j}} \bmod n \quad (5)$$

where $1 \leq k \leq n$, $k \neq i$

Each party in the group then generates the group key K as follows

P_1 generates, $K = (X_k)^{K_1} \bmod n$

$$= g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (6)$$

P_2 generates, $K = (X_k)^{K_2} \bmod n$

$$= g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (7)$$

P_3 generates, $K = (X_k)^{K_3} \bmod n$

$$= g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (8)$$

.....

.....

.....

P_{n-1}generates, $K = (X_k)^{K_{n-1}} \bmod n$

$$= g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (9)$$

P_n generates, $K = (X_k)^{K_n} \bmod n$

$$= g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (10)$$

Since the group controller knows all the two-party shared keys, it also generates the group key using ,

$$K = g^{K_1 K_2 K_3 \ldots K_{n-1} K_n} \bmod n \quad (11)$$

The proposed Group-DH mainly consists of two steps as summarized below,

1.2 Group-DH technique

Step 1: A member acts as a group controller and forms a two-party group with the remaining group members. Each group individually generates a DH-style key using DH technique.

Step 2: The group controller generates (n-1) public keys by raising the exponent of g with the product of (n-2) shared keys at a time and sends to the corresponding group members. On receiving, each member raises the exponent
With its own shared key and generates the group key.

2. Proposed m-party key distribution protocol (Improved group DH):

Let $P_1, P_2, \ldots, P_i \ldots, P_{m-1}, P_m$ be the group members and let P_i ($1 \leq i \leq m$) acts as a group controller.

Initially "P_i" itself forms a two-party group with each of the remaining group members, and produces
(n-1) two-party groups.

P_i selects a private key x_i and generates a public key

$$X_i = g^{x_i} \bmod n \quad (12)$$

and broad cast to the remaining group members.Also each group member P_j, where $j \neq i$ also assumes a private key and generates a public key as

$$X_j = g^{x_j} \bmod n \quad (13)$$

Where x_j is the private key of p_j and $1 \leq j \leq n$, $j \neq i$.

where x_j is the private key **of** P_j and $1 \leq j \leq n$, $j \neq i$.
Each P_j then transmits X_j to the group controller, P_i. After exchanging the public keys, each member similar to the basic DH generates a unique shared key, K_i with group controller as

$$K_i = X_i^{x_j} \bmod n = g^{x_i x_j} \bmod n \quad (14)$$

Similarly, P_i generates the same shared key, using

$$K_i = X_j^{x_i} \bmod n = g^{x_j x_i} \bmod n \quad (15)$$

It Actually generates (n-1) shared key K_i*'s* for $1 \leq j \leq m$ and $j \neq i$ for *(n-1)* parties respectively.

To produce a single group key first group controllers computes the following $Z_l's$,encrypt with $K_l's$ respectively and send to $P_l's$ respectively, for $1 \leq l \leq m$,$l \neq i$.

After receiving each P_l decrypts with their key and computes the group key k as follows.

$$Z_l = \prod_{j \neq l} K_i \bmod n, \quad (16)$$

$$where\ 1 \leq l \leq m, l \neq i, j \neq l$$

Each party in the group then generates the group key 'k 'as follows

P_1Generates,

$$K = Z_1 \times K_1 \bmod n = \prod_{j \neq i} K_i \bmod n \quad (17)$$

P_2Generates,

$$K = Z_2 \times K_2 \bmod n = \prod_{j \neq i} K_i \bmod n \quad (18)$$

.....

.....

.....

. . .

P_{m-1} Generates,

$$K = Z_{m-1} \times K_{m-1} \bmod n = \prod_{j \neq i} K_i \bmod n \quad (19)$$

P_m Generates,

$$K = Z_m \times K_m \bmod n = \prod_{j \neq i} K_i \bmod n \quad (20)$$

Since the group controller knows all the two party shared keys it also generates the group key using

$$K = \prod_{j \neq i} K_i \bmod n \quad (21)$$

3. COMPARATIVE ANALYSIS OF GROUP KEYDISTRIBUTION PROTOCOLS

protocols	Number of rounds	Number of messages	Total no of exponential operations
G.D. H-3	n+1	2n-1	5n-6
GROUP DH	n+1	n+1	2n
Improved Group DH	n+1	n+1	n

4. SECURITY OF THE GROUP-DH TECHNIQUE

We consider that the DH technique is secured as it supports strong DDH assumption. We can claim that the improved Group-DH that extends DH to multi-party system must be secured. Consider the following theorem.

4.1 Theorem

The group key derived in the application of Group-DH is indistinguishable in polynomial time from random numbers.

Proof:

Each of the two-party shared key generated in Step 1 of Group-DH is secured, because it uses DH protocol that supports DDH assumption. That is, all the two-party shared keys exchanged in Step 1 are indistinguishable from random numbers in polynomial time. Also the generation of the public keys made in Step 2 are product of multiple values. Note that this product is secured as it is obtained by multiplying the secured shared keys generated in Step 1 and also we are using encrypted connection between group controller and remaining members. In the final round, each group member multiplies public key by its own shared key. Since the Group-DH imitates the basic DH and supports DDH assumption, the group key generated by Group-DH is indistinguishable from random numbers in polynomial time, and thus secured.

5. CONCLUSION

We introduced a protocol to generate a multi-party key for large static groups. Although the technique is proposed for a static group, it may be easily extended for large dynamic groups. It uses simpler steps and needs comparatively less communication and computation costs, and it works for large groups without delays in key generation therefore may be useful for practical applications.

6. ACKNOWLEDGMENTS

Author would like to thank the Management of S.V.K.P. and Dr. K.S.R. Arts and Science College, for financial support. On the personal front, the author is grateful to her family members and especially Dr.K.R.Raju for his support and motivation in doing research work.

7. REFERENCES

[1] Whit Diffie and Martin Hellman. New Directions In Cryptography. IEEE Transactions on Information Theory, IT-22(6):644-654, November 1976.

[2] D. G. Steer , L. Strawczynski , W. Diffie , M. Wiener, A secure audio teleconference system, Proceedings on Advances in cryptology, p.520-528, February 1990, Santa Barbara, California, United States

[3] I. Ingemarsson, D. Tang, and C. Wong. A Conference Key Distribution System. IEEE Transactions on Ir,~formation Theory, 28(5):714-720, September 1982.

[4] Hugh Harney, Carl Muckenhirn, and Thomas Rivers. Group Key Management Protocol (GKMP) Architecture. INTERNEToDRAFT, September 1994.

[5] Yi Mu, Yuliang Zheng, and Yan-Xia Lin. Quantum Conference Key Distribution Systems. Technical Report 94- 6, University of Wollongong, NSW, Australia, 1994.

[6] Edward Zuk, Remarks on "The Design of a Conference Key Distribution System", Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology, p.467-468, December 13-16, 1992

[7] Michael K.Just. Methods Of Multi-party Cryptographic Key Establishment. Master's thesis, Ottwa Carb~ton Institute for Computer Science, Caleton University, Ottawa, Ontario, August 1994.

[8] Tzonelih Hwang, Cryptosystem for group oriented cryptography, Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology, p.352-360, February 1991, Aarhus, Denmark

[9] M. Burmester and Y. Desmedt. A Secure And Efficient Conference Key Distribution System. In I.B. Damgard, editor, Advances in Cryptology- EUROCRYPT '94, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1994.

[10] C.P. Schnorr. Efficient Signature Generation By Smart Cards. Journal of Cryptology, 4(3):161-174, 1991.

[11] CORPORATE NIST, The digital signature standard, Communications of the ACM, v.35 n.7, p.36-40, July 1992 [doi>10.1145/129902.129904]

[12] Stefan A. Brands, An Efficient Off-line Electronic Cash System Based On The Representation Problem., CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, 1993

[13] Bruce Schneier, Applied cryptography (2nd ed.): protocols, algorithms, and source code in C, John Wiley & Sons, Inc., New York, NY, 1995

[14] Michael Steiner , Gene Tsudik , Michael Waidner, Refinement and extension of encrypted key exchange, ACM SIGOPS Operating Systems Review, v.29 n.3, p.22-30, July 1995 [doi>10.1145/206826.206834

[15] T. Matsumoto, Y. Takashima, H. Imai. A Method Of Generating Secret Data Common To All Members Of A Specified Group. IECE Technical Report IT85-3~, September 1985.

[16] Micheal K. Reiter, A Secure Group Membership Protocol, Proceedings of the 1994 IEEE Symposium on Security and Privacy, p.176, May 16-18, 1994.

8. AUTHORS PROFILE:

Vankamamidi Srinivasa Naresh is currently working as aDirector, for the Post Graduate Department of Computer Science Courses in **S.V.K.P. and Dr. K.S.R. Arts and Science College**.He obtained an M.Sc. in Mathematics from Andhra University, an M.Phil. in Mathematics fromMadurai Kamaraj University and an M.Tech in ComputerScience and Engineering from J.N.T. University-Hyderabad. He is also a recipient of U.G.C.-C.S.I.R.JUNIOR RESEARCH FELLOSHIP and cleared NET for Lecturership

Nistala V.E.S. Murthy is currently working as a Professor in the department of Computer Science and Systems Engineering of Andhra University, Visakhapatnam. He developed f-Set Theory –wherein f-maps exists between fuzzy sets with truth values in *different* complete lattices, generalizing L-fuzzy set Theory of Goguen which generalized the [0,1]-fuzzy set theory of Zadeh, the Father of Fuzzy Set Theories. He also published papers on Representation of various Fuzzy Mathematical (Sub) structures in terms of their appropriate crisp cousins.