Practical and Provably-Secure Multicasting over High-Delay Networks*

Junghyun Nam[1], Hyunjue Kim[1], Seungjoo Kim[1], Dongho Won[1], and Hyungkyu Yang[2]

[1] School of Information and Communication Engineering,
Sungkyunkwan University, Suwon-si,
Gyeonggi-do 440-746, Korea
{jhnam, hjkim, dhwon}@dosan.skku.ac.kr, skim@ece.skku.ac.kr

[2] Department of Computer Engineering, Kangnam University, Yongin-si,
Gyeonggi-do 449-702, Korea
hkyang@kangnam.ac.kr

Abstract. This paper considers the problem of authenticated key exchange in a dynamic group in which members join and leave the group in an arbitrary fashion. A group key exchange scheme for such a dynamic group is designed to minimize the cost of the rekeying operations associated with group updates. Although a number of schemes have attempted for many years to address this problem, all provably-secure schemes are inadequate in dealing with a dynamic group where group members are spread across a wide area network; their communication overhead for group rekeying is significant in terms of the number of communication rounds or the number of messages, both of which are recognized as the dominant factors that severely slow down group key exchange over a wide area network. In this paper, we propose an efficient key exchange scheme for this scenario and prove its security against an active adversary under the factoring assumption. The proposed scheme requires only a constant number of rounds while achieving low message complexity.

1 Introduction

A group key exchange scheme is designed to allow a group of parties communicating over an insecure public network like the Internet to establish a shared secret value called a session key. This group session key is typically used to facilitate standard security services, such as authentication, confidentiality, and data integrity, in various group-oriented applications like e.g. collaborative computing, audio/video conferencing, and distributed database. In other words, the essential goal of group key exchange protocols is to efficiently implement secure

* Seungjoo Kim is the corresponding author for this paper. This work was supported by the University IT Research Center Project funded by the Korean Ministry of Information and Communication.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 493–501, 2005.

group communication channels over untrusted, open networks. The basic security requirement for a group key exchange scheme to achieve is the property referred to as (implicit) key authentication, whereby each member is assured that no one except the intended group members can obtain any information about the value of the session key. Therefore, the design of an efficient group key exchange scheme with key authentication is fundamental to network security and has recently received much attention as a consequence of the increased popularity of group-oriented applications [3, 16, 9, 14, 6, 15].

In this paper we focus on the problem of authenticated key exchange in a dynamic group, where current members may leave the group and new members may join the group at any time in an arbitrary manner. A group key exchange scheme for such a dynamic group must ensure that the session key is updated upon every membership change, so that subsequent communication sessions are protected from leaving members and previous communication sessions are protected from joining members. Although this can be achieved by running any authenticated group key exchange protocol from scratch whenever group membership changes, better handling of this dynamic membership problem has emerged as an important research goal toward efficient, scalable group rekeying [1, 7, 8, 12, 13, 16].

1.1 Related Work

In [9, 7, 8], Bresson et al. present the first formal security model for group key exchange, which is based on the work of Bellare et al. [4], and provide the first provably-secure protocols in this model. The initial work [9] assumes that group membership is static, whereas later works [7, 8] focus on the dynamic case. However, one obvious drawback of their scheme is that in case of initial group formation, its round complexity is linear in the number of users in the group. Moreover, the simultaneous joining of multiple users also takes a linear number of rounds with respect to the number of new users. Therefore, as group size grows large, this scheme becomes impractical particularly in wide area networks where the delays associated with communication are expected to dominate the cost of a group key exchange scheme.

Very recently, Katz and Yung [14] have presented a constant-round protocol which achieves both provable security and forward secrecy. This protocol, in its basic form, is based on the work of Burmester and Desmedt [10], and thus no efficiency gain over the Burmester-Desmedt protocol has been accompanied by its provable security. Indeed, this protocol adds one more round of n broadcasts for provable security, requiring in total three rounds of n broadcasts. Such a large number of message exchanges in one round is known as another negative factor that severely slows down group key exchange protocols in a wide area network setting. Furthermore, this protocol has to restart anew in the presence of any group membership change, because there is no known method to handle dynamic membership more efficiently for this protocol.

Most recently, in [6] Boyd and Nieto introduce another group key exchange protocol which is provably secure in the random oracle model [5] and requires only a single round of communication to complete key exchange. But unfortunately, this protocol does not achieve forward secrecy even if its round complexity is optimal.

1.2 Our Contribution

The unsatisfactory situation described above has prompted this work aimed at designing an efficient and provably-secure key exchange scheme for a dynamic group where users communicate over a high-delay network environment. We provide a rigorous proof of security in the model of Bresson et al. [9, 7, 8] in which an adversary controls all communication flows in the network. The concrete security reduction we exhibit in the ideal hash model is tight; breaking the semantic security of our scheme almost always leads to solving the well-established factoring problem, provided that the signature scheme used is existentially unforgeable. Our group key exchange scheme also provides perfect forward secrecy. Namely, disclosure of long-term secret keys does not compromise the security of previously established session keys.

In wide area network environments, the main source of delay is not the computational time needed for cryptographic operations, but the communication time spent in the network. Moreover, the power of computers continues to increase at a rapid pace. We refer the reader to the literature [2, 13] for detailed discussions of comparison between the communication latency in wide area networks and the computation time for modular exponentiation. As the experiment results of [2] also indicate, it is widely accepted that the number of communication rounds and the number of exchanged messages are two most important factors for efficient key exchange over a wide area network.

Table 1. Complexity comparison among group key exchange schemes that achieve both provable security and forward secrecy

		Communication				Computation
		Rounds	Messages	Unicast	Broadcast	Exponentiations
[7]	IKE	$n^{1)}$	n	$n-1$	1	$O(n^2)$
	Join	$j+1$	$j+1$	$j^{2)}$	1	$O(jn)$
	Leave	1	1		1	$O(n)$
[14]		3	$3n$		$3n$	$O(n)+O(n^2 \log n)^{3)}$
Here	IKE	2	n	$n-1$	1	$O(n)$
	Join	2	$j+1$	j	1	$O(n)$
	Leave	1	1		1	$O(n)$

IKE: Initial Key Exchange

1) The number of users in a newly updated group

2) The number of joining users

3) $O(n^2 \log n)$: the number of modular multiplications

Table 1 compares the efficiency of our scheme given in Section 3 with other provably-secure schemes that provide forward secrecy [7, 14]. As for computational costs, the table lists the total amount of computation that needs to be done by users. As shown in the table, the scheme of [7] requires n communication rounds for initial key exchange which occurs at the time of group genesis, and j communication rounds for the rekeying operation that follows the joining of j new users. The protocol of [14], as already mentioned, requires n broadcast messages to be sent in each of three rounds, both for initial key exchange and for every group rekeying operation. In contrast, our scheme takes at most 2 communication rounds while maintaining low message complexity, in any of the three cases. Therefore, it is straightforward to see that our dynamic group key exchange scheme is well suited for networking environments with high communication latency. In particular, due to its computational asymmetry, our scheme is best suited for unbalanced networks consisting of mobile hosts with restricted computational resources and stationary hosts with relatively high computational capabilities.

2 Security Definitions

In this section, we first define what it means to securely distribute a session key within the security model given above and then explore the underlying assumptions on which the security of our scheme rests.

Authenticated Group Key Exchange. The security of an authenticated group key exchange scheme P is defined in the following context. The adversary $\mathcal{A}$, equipped with all the queries described in the security model, executes the protocols IKE1, LP1, and JP1 as many times as she wishes in an arbitrary order, of course, with IKE1 being the first one executed. During executions of the protocols, the adversary $\mathcal{A}$, at any time, asks a Test query to a fresh user, gets back an ℓ-bit string as the response to this query, and at some later point in time, outputs a bit b' as a guess for the secret bit b. Let $\mathsf{Good\text{-}Guess}$ be the event that the adversary $\mathcal{A}$ correctly guesses the bit b, i.e., the event that $b' = b$. Then we define the advantage of $\mathcal{A}$ in attacking P as

$$\mathsf{Adv}_P^{\mathcal{A}}(k) = 2 \cdot \Pr[\mathsf{Good\text{-}Guess}] - 1,$$

where k is the security parameter. We say that a group key exchange scheme P is secure if $\mathsf{Adv}_P^{\mathcal{A}}(k)$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$.

Secure Signature Schemes. We review here the standard definition of a digital signature scheme. A digital signature scheme $\Gamma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ is defined by the following triple of algorithms:

- A *probabilistic key generation algorithm* $\mathcal{G}$, on input 1^k, outputs a pair of matching public and private keys (PK, SK).
- A *signing algorithm* $\mathcal{S}$ is a (possibly probabilistic) polynomial time algorithm that, given a message m and a key pair (PK, SK) as inputs, outputs a signature σ of m.

– A *verification algorithm* $\mathcal{V}$ is a (usually deterministic) polynomial time algorithm that on input (m, σ, PK), outputs 1 if σ is a valid signature of the message m with respect to PK, and 0 otherwise.

We denote by $\mathsf{Succ}_{\Gamma}^{\mathcal{A}}(k)$ the probability of an adversary $\mathcal{A}$ succeeding with an existential forgery under adaptive chosen message attack [11]. We say that a signature scheme Γ is secure if $\mathsf{Succ}_{\Gamma}^{\mathcal{A}}(k)$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$. We denote by $\mathsf{Succ}_{\Gamma}(t)$ the maximum value of $\mathsf{Succ}_{\Gamma}^{\mathcal{A}}(k)$ over all adversaries $\mathcal{A}$ running in time at most t.

Factoring Assumption. Let $\mathcal{FIG}$ be a factoring instance generator that on input 1^k, runs in time polynomial in k and outputs a $2k$-bit integer $N = p \cdot q$, where p and q are two random distinct k-bit primes such that $p \equiv q \equiv 3 \pmod 4$. Then, we define $\mathsf{Succ}_{N}^{\mathcal{A}}(k)$ as the advantage of adversary $\mathcal{A}$ in factoring $N = p \cdot q$ chosen from $\mathcal{FIG}(1^k)$. Namely,

$$\mathsf{Succ}_{N}^{\mathcal{A}}(k) = \Pr[\mathcal{A}(N) \in \{p, q\} \mid N(= pq) \longleftarrow \mathcal{FIG}(1^k)].$$

We say that $\mathcal{FIG}$ satisfies the factoring assumption if for all sufficiently large k, $\mathsf{Succ}_{N}^{\mathcal{A}}(k)$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$. Similarly as before, we denote by $\mathsf{Succ}_{N}(t)$ the maximum value of $\mathsf{Succ}_{N}^{\mathcal{A}}(k)$ over all adversaries $\mathcal{A}$ running in time at most t.

3 The Proposed Scheme

We now present a dynamic group key exchange scheme consisting of three protocols IKE1, LP1, and JP1 for initial group formation, user leave, and user join, respectively.

Let N be any possible output of $\mathcal{FIG}(1^k)$ and let $g \neq 1$ be a quadratic residue that is chosen uniformly at random in the set of quadratic residues in $\mathbb{Z}_N^*$, where $\mathbb{Z}_N^*$ is the multiplicative group modulo N. Then, we define the finite group $\mathbb{G}$, over which we must work, to be the cyclic subgroup of $\mathbb{Z}_N^*$ generated by g. For the rest of this paper, we denote by U_c the controller in a multicast group $\mathcal{MG}$, and by $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ a hash function modelled as a random oracle in the security proof of the scheme. For simplicity, we will often omit "mod N" from expressions if no confusion arises.

3.1 Initial Key Exchange: Protocol IKE1

Assume a multicast group $\mathcal{MG} = \{U_1, U_2, \ldots, U_n\}$ of n users who wish to establish a session key by participating in protocol IKE1. Then IKE1 runs in two rounds, one with $n-1$ unicasts and the other with a single broadcast, as follows:

1. Each U_i picks a random $r_i \in [1, N]$ and computes $z_i = g^{r_i} \bmod N$. $U_i \neq U_c$ then signs $U_i \| z_i$ to obtain signature σ_i and sends $m_i = U_i \| z_i \| \sigma_i$ to the controller U_c.

2. Upon receiving each message m_i, U_c verifies the correctness of m_i and computes $y_i = z_i^{r_c} \bmod N$. After receiving all the $n-1$ messages, U_c computes Y as $Y = \prod_{i \in [1,n] \setminus \{c\}} y_i \bmod N$ if n is even, and as $Y = \prod_{i \in [1,n]} y_i \bmod N$ if n is odd. U_c also computes the set $\mathcal{T} = \{T_i \mid i \in [1,n] \setminus \{c\}\}$ where $T_i = Y \cdot y_i^{-1} \bmod N$. Let $\mathcal{Z} = \{z_i \mid i \in [1,n]\}$. Then, U_c signs $\mathcal{MG} \| \mathcal{Z} \| \mathcal{T}$ to obtain signature σ_c and broadcasts $m_c = \mathcal{MG} \| \mathcal{Z} \| \mathcal{T} \| \sigma_c$ to the entire group.
3. Upon receiving the broadcast message m_c, each $U_i \neq U_c$ verifies the correctness of m_c and computes $Y = z_c^{r_i} \cdot T_i \bmod N$. All users in $\mathcal{MG}$ compute their session key as $K = \mathcal{H}(\mathcal{T} \| Y)$, and store their random exponent r_i and the set $\mathcal{Z}$ for future use.

To take a simplified example as an illustration, consider a multicast group $\mathcal{MG} = \{U_1, U_2, \ldots, U_5\}$ and let $U_c = U_5$. Then, in IKE1, the controller U_5 receives $\{g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}\}$ from the rest of the users, and broadcasts $\mathcal{Z} = \{g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}, g^{r_5}\}$ and $\mathcal{T} = \{g^{r_5(r_2+r_3+r_4+r_5)}, g^{r_5(r_1+r_3+r_4+r_5)}, g^{r_5(r_1+r_2+r_4+r_5)}, g^{r_5(r_1+r_2+r_3+r_5)}\}$. All users in $\mathcal{MG}$ compute the same key: $K = \mathcal{H}(\mathcal{T} \| Y)$, where $Y = g^{r_5(r_1+r_2+r_3+r_4+r_5)}$.

3.2 User Leave: Protocol LP1

Assume a scenario where a set of users $\mathcal{L}$ leaves a multicast group $\mathcal{MG}_p$. Then protocol LP1 is executed to provide each user of the new multicast group $\mathcal{MG}_n = \mathcal{MG}_p \setminus \mathcal{L}$ with a new session key. Any remaining user can act as the controller in the new multicast group $\mathcal{MG}_n$. LP1 requires only one communication round with a single broadcast and it proceeds as follows:

1. U_c picks a new random $r'_c \in [1, N]$ and computes $z'_c = g^{r'_c} \bmod N$. Using r'_c, z'_c and the saved set $\mathcal{Z}$, U_c then proceeds exactly as in IKE1, except that it broadcasts $m_c = \mathcal{MG}_n \| z_c \| z'_c \| \mathcal{T} \| \sigma_c$ where z_c is the random exponential from the previous controller.
2. Upon receiving the broadcast message m_c, each $U_i \neq U_c$ verifies that: (1) $\mathcal{V}(\mathcal{MG}_n \| z_c \| z'_c \| \mathcal{T}, \sigma_c, PK_c) = 1$ and (2) the received z_c is equal to the random exponential from the previous controller. All users in $\mathcal{MG}_n$ then compute their session key as $K = \mathcal{H}(\mathcal{T} \| Y)$ and update the set $\mathcal{Z}$.

We assume that in the previous example, a set of users $\mathcal{L} = \{U_2, U_4\}$ leaves the multicast group $\mathcal{MG}_p = \{U_1, U_2, \ldots, U_5\}$ and hence the remaining users form a new multicast group $\mathcal{MG}_n = \{U_1, U_3, U_5\}$. Also assume that U_5 remains as the controller in the new multicast group $\mathcal{MG}_n$. Then U_5 chooses a new random value r'_5, and broadcasts z_5, $z'_5 = g^{r'_5}$, and $\mathcal{T} = \{g^{r'_5(r_3+r'_5)}, g^{r'_5(r_1+r'_5)}\}$. All users in $\mathcal{MG}_n$ compute the same key: $K = \mathcal{H}(\mathcal{T} \| Y)$, where $Y = g^{r'_5(r_1+r_3+r'_5)}$.

3.3 User Join: Protocol JP1

Assume a scenario in which a set of j new users $\mathcal{J}$ joins a multicast group $\mathcal{MG}_p$ to form a new multicast group $\mathcal{MG}_n = \mathcal{MG}_p \cup \mathcal{J}$. Then the join protocol JP1 is run to provide the users of $\mathcal{MG}_n$ with a session key. Any user from the previous multicast group $\mathcal{MG}_p$ can act as the controller in the new multicast group $\mathcal{MG}_n$. JP1 takes two communication rounds, one with j unicasts and the other with a single broadcast, and it proceeds as follows:

1. Each $U_i \in \mathcal{J}$ picks a random $r_i \in [1, N]$ and computes $z_i = g^{r_i} \bmod N$. $U_i \in \mathcal{J}$ then generates signature σ_i of $U_i \| z_i$, sends $m_i = U_i \| z_i \| \sigma_i$ to U_c, and stores its random r_i.
2. U_c proceeds in the usual way, choosing a new random r'_c, computing z'_c, Y, $\mathcal{T}$ and $K = \mathcal{H}(\mathcal{T} \| Y)$, updating the set $\mathcal{Z}$ with new z_i's, and then broadcasting $m_c = \mathcal{MG}_n \| z_c \| \mathcal{Z} \| \mathcal{T} \| \sigma_c$.
3. After verifying the correctness of m_c (including the verification by $U_i \in \mathcal{MG}_p \setminus \{U_c\}$ that the received z_c is equal to the random exponential from the previous controller), each $U_i \neq U_c$ proceeds as usual, computing $Y = {z'}_c^{r_i} \cdot T_i \bmod N$ and $K = \mathcal{H}(\mathcal{T} \| Y)$. All users in $\mathcal{MG}_n$ store or update the set $\mathcal{Z}$.

Consider the same example as used for LP1 and assume that a set of users $\mathcal{J} = \{U_2\}$ joins the multicast group $\mathcal{MG}_p = \{U_1, U_3, U_5\}$ to form a new multicast group $\mathcal{MG}_n = \{U_1, U_2, U_3, U_5\}$. Also assume that controller $U_c = U_5$ remains unchanged from $\mathcal{MG}_p$ to $\mathcal{MG}_n$. Then, U_5 receives $\{g^{r'_2}\}$ from the users in $\mathcal{J}$, and broadcasts z'_5, $\mathcal{Z} = \{g^{r_1}, g^{r'_2}, g^{r_3}, g^{r''_5}\}$ and $\mathcal{T} = \{g^{r''_5(r'_2+r_3)}, g^{r''_5(r_1+r_3)}, g^{r''_5(r_1+r'_2)}\}$ to the rest of the users, where r''_5 is the new random exponent of controller U_5. All users in $\mathcal{MG}_n$ compute the same key: $K = \mathcal{H}(\mathcal{T} \| Y)$, where $Y = g^{r''_5(r_1+r'_2+r_3)}$.

4 Security Result

Theorem 1. *Let the number of potential participants be bounded by a polynomial function $p_u(k)$ of the security parameter k. Let $\mathsf{Adv}_P(t, q_{se}, q_h)$ be the maximum advantage in attacking P, where the maximum is over all adversaries that run in time t, and make q_{se} Send queries and q_h random oracle queries. Then we have*

$$\mathsf{Adv}_P(t, q_{se}, q_h) \leq 2 \cdot \mathsf{Succ}_N(t') + 2p_u(k) \cdot \mathsf{Succ}_\Gamma(t''),$$

where $t' = t + O(q_{se} p_u(k) t_{exp} + q_h t_{exp})$, $t'' = t + O(q_{se} p_u(k) t_{exp})$, and t_{exp} is the time required to compute a modular exponentiation in $\mathbb{G}$.

In the following, we briefly outline the proof of Theorem 1[1]. The proof is divided into two cases: (1) the case that the adversary $\mathcal{A}$ breaks the scheme by forging a signature with respect to some user's public key, and (2) the case that $\mathcal{A}$ breaks the scheme without forging a signature. We argue by contradiction, assuming that there exists an adversary $\mathcal{A}$ who has a non-negligible advantage in

[1] The complete proof of the theorem is omitted here due to lack of space, and is given in the full version of this paper, which is available at *http://eprint.iacr.org/2004/115*.

attacking P. For the case (1), we reduce the security of scheme P to the security of the signature scheme Γ, by constructing an efficient forger $\mathcal{F}$ who given as input a public key PK and access to a signing oracle associated with this key, outputs a valid forgery with respect to PK. For the case (2), the reduction is from the factoring problem; given the adversary $\mathcal{A}$, we build an efficient factoring algorithm $\mathcal{B}$ which given as input $N = p \cdot q$ generated by $\mathcal{FIG}(1^k)$, outputs either p or q.

References

1. D.A. Agarwal, O. Chevassut, M.R. Thompson, and G. Tsudik: An Integrated Solution for Secure Group Communication in Wide-Area Networks. In Proc. of 6th IEEE Symposium on Computers and Communications, pp. 22–28, 2001.
2. Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik: On the Performance of Group Key Agreement Protocols. ACM Trans. on Information and System Security, vol.7, no.3, pp. 457–488, August 2004.
3. K. Becker, and U. Wille: Communication complexity of group key distribution. In Proc. of 5th ACM Conf. on Computer and Communications Security, pp. 1–6, 1998.
4. M. Bellare, D. Pointcheval, and P. Rogaway: Authenticated key exchange secure against dictionary attacks, Eurocrypt'00, LNCS 1807, pp. 139–155, 2000.
5. M. Bellare and P. Rogaway: Random oracles are practical: A paradigm for designing efficient protocols. In Proc. of 1st ACM Conf. on Computer and Communications Security (CCS'93), pp. 62–73, 1993.
6. C. Boyd and J.M.G. Nieto: Round-optimal contributory conference key agreement. PKC2003, LNCS 2567, pp. 161–174, 2003.
7. E. Bresson, O. Chevassut, and D. Pointcheval: Provably authenticated group Diffie-Hellman key exchange — the dynamic case. Asiacrypt'01, pp. 290–309, 2001.
8. E. Bresson, O. Chevassut, and D. Pointcheval: Dynamic group Diffie-Hellman key exchange under standard assumptions. Eurocrypt'02, pp. 321–336, 2002.
9. E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater: Provably authenticated group Diffie-Hellman key exchange. In Proc. of 8th ACM Conf. on Computer and Communications Security, pp. 255–264, 2001.
10. M. Burmester and Y. Desmedt: A secure and efficient conference key distribution system. Eurocrypt'94, LNCS 950, pp. 275–286, 1994.
11. S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal of Computing, vol.17, no.2, pp. 281–308, 1988.
12. Y. Kim, A. Perrig, and G. Tsudik: Simple and fault-tolerant key agreement for dynamic collaborative groups. In Proc. of 7th ACM Conf. on Computer and Communications Security, pp. 235–244, 2000.
13. Y. Kim, A. Perrig, and G. Tsudik: Communication-efficient group key agreement. In Proc. of International Federation for Information Processing — 16th International Conference on Information Security (IFIP SEC'01), pp. 229–244, June 2001.

14. J. Katz and M. Yung: Scalable protocols for authenticated group key exchange. Crypto'03, LNCS 2729, pp. 110–125, August 2003.
15. J. Nam, S. Cho, S. Kim, and D. Won: Simple and efficient group key agreement based on factoring. In Proc. of the 2004 International Conference on Computational Science and Its Applications (ICCSA 2004), LNCS 3043, pp. 645–654, May 2004.
16. M. Steiner, G. Tsudik, and M. Waidner: Key agreement in dynamic peer groups. IEEE Trans. on Parallel and Distrib. Syst., vol.11, no.8, pp. 769–780, August 2000.