# Secure Group Communications over Combined Wired/Wireless Networks

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

[1] School of Information and Communication Engineering,
Sungkyunkwan University, Korea
{jhnam,skim}@ece.skku.ac.kr, dhwon@simsan.skku.ac.kr

[2] Division of Industrial Engineering, Computer Science, and Electronic Engineering,
Kangnam University, Korea
hkyang@kns.kangnam.ac.kr

**Abstract.** This paper considers the fundamental problem of key agreement among a group of parties communicating over an insecure public network. Over the years, a number of solutions to this problem have been proposed with varying degrees of complexity. However, there seems to have been no previous systematic look at the growing problem of key agreement over combined wired/wireless networks, consisting of both high-performance computing machines and low-power mobile devices. In this paper we present an efficient group key agreement scheme well suited for this networking environment. Our construction is intuitively simple, and yet offers a scalable solution to the problem.

**Keywords:** Group key agreement, combined wired/wireless networks, mobile devices, DDH assumption

## 1 Introduction

A group key agreement protocol is designed to allow a group of parties communicating over an untrusted, open network to share a secret value called a *session key*. This common session key is typically used to facilitate standard security services, such as authentication, confidentiality, and data integrity, in various applications which are likely to involve a large number of users. As these group-oriented applications proliferate in modern computing environments (e.g., video conferencing, multi-player game, and replicated database), the design of an efficient group key agreement protocol has received much attention in the literature [8, 15, 3, 12, 4, 5, 13] as an important research goal. The efficiency of group key agreement protocols is measured with respect to communication complexity, as well as computational complexity. Communication complexity is quantified as both the number of rounds of communication among users and the number of messages sent/received by users, while computational complexity is mostly concerned with the number of public-key cryptography operations that users have to perform. For a group key agreement protocol to be scalable, it is of prime importance in many real-life applications that the protocol be able to run only in a constant number of communication rounds.

In this paper we consider the scenario where limited-function devices, such as PDAs and handheld computers, and general-purpose computing machines like servers and desktop computers coexist participating in the same group. When one considers the broad range of wirelessly connected mobile devices used today, it is clear that integrating such network-enabled devices into secure group communication systems is timely and will be increasingly important. Although mobile devices represent an already large and growing percentage of the computing population, security is still a major gating factor for their full adoption. Despite all the work conducted over many decades, the implementation of strong protection in a mobile environment is non-trivial [2]. Security solutions targeted for more traditional networks are often not directly applicable to wireless networks due to a marked difference in computing resources between mobile devices and stationary computers.

Indeed, most of previous group key agreement protocols are not well suited for networking environments similar to our setting. Even though some constant-round protocols have been proposed [8, 12, 5], they are still too costly to be practical for applications involving mobile devices with limited computing resources. The reason for this is that these protocols are fully symmetric and therefore, as group size grows, the workload of every user also increases substantially, imposing an unfair, excessive burden on small mobile devices. Other constant-round protocols [4, 6], while they require only a fixed amount of computation for all but one group member, do not provide perfect forward secrecy [10]; i.e., earlier session keys are compromised by loss of some underlying information at the present time. Furthermore, in these protocols one special user must perform $O(n)$ public-key cryptography operations in a group of size $n$, being a significant performance bottleneck in a large group setting.

In this work we focus on *contributory* key agreement protocols in which the session key is derived as a function of contributions provided by all parties. In contributory key agreement protocols, a correctly behaving party is assured that as long as his contribution is chosen at random, even a coalition of all other parties will not be able to have any means of controlling the final value of the session key. Therefore, contributory key agreement protocols are fairer and more secure than key transport protocols. Thus, it is often recommended to use contributory key agreement to prevent some parties having any kind of advantage over the others [1]. Moreover, most key transport protocols [16, 14], while they focus on minimizing the cost of the rekeying operations associated with group updates, lack at least one of the important security properties: perfect forward secrecy or known key security.

Our main contribution is an efficient constant-round scheme for contributory group key agreement over combined wired/wireless networks, consisting of arbitrary numbers of mobile devices and stationary high-performance computers. While a number of problems related to group key agreement have been tackled and solved over the past years, there seems to have been no previous systematic look at the growing problem of group key agreement in this networking environment. In order to generalize the problem, we broadly divide all the users of the network into two groups, namely, users that have sufficient computational capabilities and users that have relatively low computing resources. By evenly spreading most of workload across high power users, we avoid any potential performance bottleneck of the system while keeping the computational cost of low power users at minimal. Our group key agreement scheme is also very efficient in terms of communication complexity which includes both round and message complexities. Without respect to the number of users, our scheme requires only a constant number of communication rounds and furthermore achieves optimal message complexity [3]. Communication complexity is especially relevant in today's computing environments where the rapid increase in computation power of computers exposed high network delay and congestion as a major bottleneck in group key agreement schemes.

The remainder of this paper is organized as follows. First, we review some of the most well-known protocols in the next section. Then, we set up some notation and assumptions in Section 3, and propose our group key agreement scheme in Section 4. Finally, we discuss the efficiency and the security of the proposed scheme in Section 5 and Section 6, respectively.

## 2 Related Work

This section describes some of previous works including all the constant-round protocols published up to date. The original idea of extending the 2-party Diffie-Hellman scheme [9] to the multi-party setting dates back to the classical paper of Ingemarsson et al. [11], and is followed by many works [8, 15, 3] offering various levels of complexity. But, only recently have Bresson et al. [7] proposed the first group key agreement protocol proven secure in a well-defined security model. This provably-secure protocol is based on one of the protocols of Steiner et al. [15] and requires $n$ communication rounds to establish a session key among a group of $n$ parties. Therefore, as group size grows large, this protocol becomes impractical particularly in wide area networks where the delays associated with communication dominate the cost of group key agreement protocols.

**Fully Symmetric Protocols.** Using the security model of Bresson et al. [7], Katz and Yung [12] have recently proposed the first constant-round and provably-secure protocol for group key agreement. More precisely, they provide a formal proof of security for the two-round protocol of Burmester and Desmedt [8], and introduce a one-round compiler that transforms any group key agreement protocol secure against a passive adversary into one that is secure against an active adversary. While these protocols [8, 12] are very efficient in general, they are not well suited for applications deployed over a combined wired/wireless network. Due to the full symmetry of the protocols, each mobile device has to receive $O(n)$ messages, and perform 3 modular exponentiations, $O(n \log n)$ modular multiplications, $O(n)$ signature verifications, and 2 signature generations. Most recently, in [5], Bresson and Catalano have introduced another fully-symmetric protocol which requires two rounds of communication. Interestingly, unlike previous approaches, they construct the protocol by combining the properties of the ElGamal encryption scheme with standard secret sharing techniques. However, with increasing number of users, the complexity of the protocol becomes beyond the capabilities of small mobile devices.

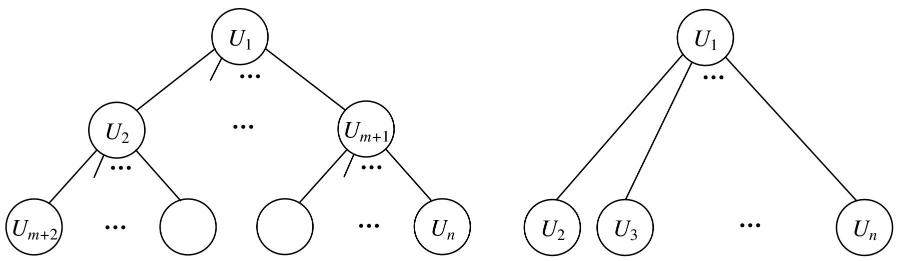
**Extremely Asymmetric Protocols.** In [4], Boyd and Nieto have presented the first group key agreement protocol that can be completed in a single round of communication. But unfortunately, this protocol does not achieve perfect forward secrecy even if its round complexity is optimal; it still remains an open problem to find a one-round group key agreement protocol providing forward secrecy. In 2003, another constant-round protocol that does not achieve forward secrecy has been offered by Bresson et al. [6]. This protocol provides an efficient method to agree on a session key between a gateway and a cluster of mobile devices. However, in common with the protocol of Boyd and Nieto [4], this protocol suffers from extreme asymmetry in the sense that one distinct user performs $O(n)$ computations whereas the other users perform only $O(1)$ computations. Consequently, none of previous research addresses well the problem of group key agreement over combined wired/wireless networks.

## 3 Protocol Preliminaries

We fix a nonempty set $\mathcal{U}$ of $n$ users who wish to agree on a common session key by participating in a group key agreement protocol. Let $\mathcal{U} = \mathcal{S} \cup \mathcal{R}$, where $\mathcal{S} = \{U_1, \ldots, U_{n_h}\}$ is the nonempty set of users that have sufficient computational capabilities and $\mathcal{R} = \{U_{n_h+1}, \ldots, U_n\}$ is the set of users that have relatively restricted computing resources. As depicted in Fig. 1(a), the users are arranged in a tree structure with height 2 according to their computing power. All users in $\mathcal{R}$ are at leaves in the tree while the users in $\mathcal{S}$ could be at any level in the hierarchy from 0 to 2. Let $n_l$ denote the cardinality of $\mathcal{R}$ (i.e., $n = n_h + n_l$). Given $n_h$ and $n_l$, the number of users at level 1, $m$, is determined as follows, aiming to minimize the maximum amount of computation that one has to perform during an execution of the protocol.

$$m = \begin{cases} 0 & \text{if } n_h = 1 \text{ or } n_h = 2 \\ n_h - 1 & \text{if } n_l \geq (n_h - 1)(n_h - 2) \\ k & \text{otherwise,} \end{cases}$$

where $k$ is the largest positive integer such that $k^2 \leq n - 1$. Fig. 1(b) shows one extreme case where $m = 0$ (i.e., $n_h = 1$ or $n_h = 2$), and thus, the users are organized into an $(n-1)$-ary tree with height 1.




(a) $n_h > 2$ (b) $n_h = 1$ or $n_h = 2$

**Fig. 1.** $\mathcal{U} = \mathcal{S} \cup \mathcal{R}$, $\mathcal{S} = \{U_1, \ldots, U_{n_h}\}$, $\mathcal{R} = \{U_{n_h+1}, \ldots, U_n\}$

In the next section, we first construct a two-round protocol for the extreme case $n_h \leq 2$ and then show that an efficient three-round protocol for the case $n_h > 2$ can be constructed by generalizing the idea of the two-round protocol. Due to lack of space, we focus on security against passive adversaries and assume all messages are digitally signed by their source in a way that the signatures cannot be forged.

To simplify the descriptions of the protocols, we divide the set $\mathcal{U}$ into three disjoint subsets $\mathcal{L}_0$, $\mathcal{L}_1$ and $\mathcal{L}_2$ which denote the sets of users at level 0, 1 and 2, respectively. We assume that all users know the structure of the tree and their position within the tree. Furthermore, the finite cyclic group $\mathbb{G} = \langle g \rangle$ of $\ell$-bit prime order $q$ is assumed to be known in advance. There is also a one-way hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\ell$ modelled as a random oracle in the security proof.

## 4 The Proposed Scheme

This section introduces new constant-round protocols for group key agreement, which take advantage of the difference in computing power between users.

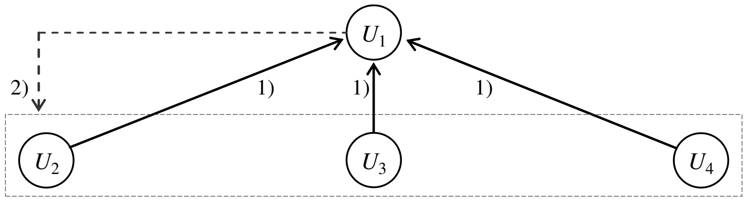
### 4.1 Basic Protocol

Consider the case $n_h \leq 2$. The protocol for this case, on input three sets $\mathcal{L}_0 = \{U_1\}$, $\mathcal{L}_1 = \{U_2, \ldots, U_n\}$, and $\mathcal{L}_2 = \emptyset$, is performed in two communication rounds, the first with $n-1$ unicasts and the second with a single broadcast, as follows (see Fig. 2 for an example):

**Round 1.** Each user $U_i \in \mathcal{L}_1$ chooses a random $r_i \in \mathbb{Z}_q$ and computes $z_i = g^{r_i}$, and sends $z_i$ to its parent $U_1$, who chooses random $s, r_1 \in \mathbb{Z}_q$ and computes $w = g^s$ and $x_1 = g^{sr_1}$.

**Round 2.** User $U_1$ computes $x_i = z_i^s$ upon receiving each $z_i$. After computing $X = \prod_{i \in [1,n]} x_i$ and the set $\mathcal{Y} = \{y_i \mid i \in [2,n]\}$, where $y_i = X \cdot x_i^{-1}$, user $U_1$ broadcasts $w \| \mathcal{Y}$ to its children.

**Key computation.** Upon receiving the broadcast, each user $U_i \in \mathcal{L}_1$ computes $X = y_i \cdot w^{r_i}$. All users in $\mathcal{U}$ compute their session key as $K = H(\mathcal{Y}, X)$.




1) Round 1: $g^{r_2}, g^{r_3}, g^{r_4}$.
2) Round 2: $w = g^s, \mathcal{Y} = \{g^{s(r_1+r_3+r_4)}, g^{s(r_1+r_2+r_4)}, g^{s(r_1+r_2+r_3)}\}$.

**Fig. 2.** An execution of the basic protocol with $\mathcal{U} = \{U_1, U_2, U_3, U_4\}$

### 4.2 Generalized Protocol

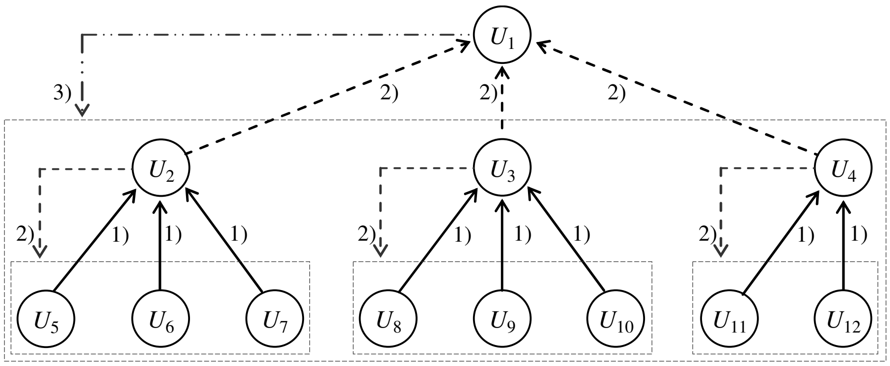
This subsection presents our main construction which uses as a basic building block the two-round protocol described above. The idea is to distribute the users into $m$ subgroups and to run the basic protocol for each subgroup. After having derived a shared secret value, each subgroup participates again in the basic protocol as a single entity to generate the final group key. Each parent $U_p \in \mathcal{L}_1$ forms a subgroup with its children (see Fig. 1(a)) and takes charge of the central control in that subgroup. We denote by $\mathcal{I}_p$ the set of indices of the children of user $U_p$. Now the users in three nonempty sets, $\mathcal{L}_0 = \{U_1\}$, $\mathcal{L}_1 = \{U_2, \ldots, U_{m+1}\}$ and $\mathcal{L}_2 = \{U_{m+2}, \ldots, U_n\}$, agree on a common session key as follows (see also Fig. 3):

**Round 1.** Each user $U_i \in \mathcal{L}_2$ chooses a random $r_i \in \mathbb{Z}_q$ and computes $z_i = g^{r_i}$, and sends $z_i$ to its parent. The other users (i.e., the users with children) select two random values; user $U_1$ chooses random $s_1, k_1 \in \mathbb{Z}_q$ and computes $w_1 = g^{s_1}$ and $\hat{x}_1 = g^{s_1 k_1}$, and user $U_p \in \mathcal{L}_1$ chooses random $s_p, r_p \in \mathbb{Z}_q$ and computes $w_p = g^{s_p}$ and $x_{pp} = g^{s_p r_p}$.

**Round 2.** Each user $U_p \in \mathcal{L}_1$, upon receiving each message $z_i$ for $i \in \mathcal{I}_p$, computes $x_{pi} = z_i^{s_p}$. After computing $X_p = \prod_{i \in \mathcal{I}_p \cup \{p\}} x_{pi}$, the set $\mathcal{Y}_p = \{Y_i \mid i \in \mathcal{I}_p\}$, where $Y_i = X_p \cdot x_{pi}^{-1}$, the subgroup key $k_p = H(\mathcal{Y}_p, X_p)$, and $\hat{z}_p = g^{k_p}$, user $U_p$ broadcasts $m_p = \hat{z}_p \| w_p \| \mathcal{Y}_p$.

**Round 3.** The user $U_1 \in \mathcal{L}_0$, upon receiving each message $m_p$ for $p \in [2, m+1]$, computes $\hat{x}_p = \hat{z}_p^{s_1}$. After computing $X_1 = \prod_{p \in [1, m+1]} \hat{x}_p$, $\mathcal{Y}_1 = \{\hat{Y}_p \mid p \in [2, m+1]\}$, where $\hat{Y}_p = X_1 \cdot \hat{x}_p^{-1}$, user $U_1$ broadcasts $w_1 \| \mathcal{Y}_1$.

**Key computation.** Now for all $p \in [2, m+1]$ and all $i \in \mathcal{I}_p$, user $U_i$ is able to generate the session key $K$; first $U_i$ calculates $k_p = H(\mathcal{Y}_p, X_p)$ with $X_p = Y_i \cdot w_p^{z_i}$ and then $K = H(\mathcal{Y}_1, X_1)$ with $X_1 = \hat{Y}_p \cdot w_1^{k_p}$.




1) Round 1: $z_5, \ldots, z_{12}$. 2) Round 2: $\hat{z}_2 \| w_2 \| \mathcal{Y}_2, \ldots, \hat{z}_4 \| w_4 \| \mathcal{Y}_4$. 3) Round 3: $w_1 \| \mathcal{Y}_1$.

**Fig. 3.** An execution of the generalized protocol with $\mathcal{U} = \{U_1, \ldots, U_{12}\}$

## 5 Efficiency

To the best of our knowledge, the protocol of Burmester and Desmedt [8] (often called the BD protocol) is the most efficient one among forward-secure group key agreement protocols published up to date. Therefore, in Table 1 we compare the efficiency of our protocols with the BD protocol. As for computational costs, the table lists the amount of computation that each user has to perform.

The protocols proposed in this paper are very efficient in terms of both round and message complexities. In particular, both the two- and three-round protocols achieve *optimal* message complexity, requiring only $n$ messages (see Theorem 2 of [3]). Our group key agreement protocols are also very efficient in terms of the computational cost of mobile devices. If precomputations are possible, all the exponentiations in the first round of the protocols can be performed off-line and thus, only one or two exponentiations per mobile device is required to be done on-line. Furthermore, the three-round protocol avoids any potential performance bottleneck by distributing computation among the high power users; the maximum computation rate per user is bounded by $O(\sqrt{n})$ with the reasonable assumption that the number of high power users is at least $\sqrt{n_l}$.

On the other hand, in the BD protocol, all users behave in a completely symmetric manner; each user broadcasts one message per round, and performs 3 modular exponentiations and $O(n \log n)$ modular multiplications. While this protocol takes only two communication rounds, the full symmetry negatively impacts on the overall performance of the protocol involving mobile devices. The number of messages received by each mobile device is $O(n)$ compared to $O(1)$ in our protocols. This implies that in the BD protocol, all users including mobile users have to perform $O(n)$ signature verifications. Moreover, the number of modular multiplications per user increases rapidly as group size grows.

We summarize as follows: in situations where users with equal computational capabilities communicate over a broadcast network, the fully-symmetric protocol of Burmester and Desmedt might be more favorable than our protocols which, in contrast, are well suited for more realistic settings where users with asymmetric computing powers are spread across a wide area network.

**Table 1.** Complexity comparison with the protocol of Burmester and Desmedt [8]

| | Communication | | | Computation | |
|---|---|---|---|---|---|
| | Rounds | Unicasts | Broadcasts | Low Power User | High Power User |
| BD | 2 | | $2n$ | 3E + $O(n)$V + $O(n \log n)$M | |
| Basic | 2 | $n-1$ | 1 | 2E + 1V | $O(n)$E + $O(n)$V |
| Generalized | 3 | $n-m-1$ | $m+1$ | 3E + 2V | $O(\sqrt{n})$E + $O(\sqrt{n})$V |

E: Exponentiation, V: Verification, M: Multiplication

## 6 Security of the Protocols

The main new building block of our scheme is the two-round protocol for the case $n_h \leq 2$. Hence, we restrict our discussion to proving that the security of the two-round protocol is based on the well-studied Decisional Diffie-Hellman (DDH) assumption; yet the security of the three-round protocol can be proved in a similar way by using the random self-reducibility of the DDH problem, and its proof will be given in the full version of this paper.

Before describing the details of the proof, let us first define $\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t)$ as the maximum value, over all distinguishers $\mathcal{D}$ running in time at most $t$, of:

$$\Big|\Pr[\mathcal{D}(g, g^x, g^y, g^{xy}) = 1 \mid x, y \leftarrow \mathbb{Z}_q] - \Pr[\mathcal{D}(g, g^x, g^y, g^z) = 1 \mid x, y, z \leftarrow \mathbb{Z}_q]\Big|.$$

Now we consider the following two distributions:

$$\mathsf{Real} = \left\{ (\mathsf{T}, K) \;\middle|\; \begin{array}{l} r_1, \ldots, r_n, s \in_R \mathbb{Z}_q; \\ z_1 = g^{r_1}, \ldots, z_n = g^{r_n}, w = g^s; \\ x_1 = g^{sr_1}, \ldots, x_n = g^{sr_n}; \\ X = x_1 \cdots x_n; \\ Y_2 = X \cdot x_2^{-1}, \ldots, Y_n = X \cdot x_n^{-1} \end{array} \right\},$$

$$\mathsf{Fake} = \left\{ (\mathsf{T}, K) \;\middle|\; \begin{array}{l} r_1, \ldots, r_n, s, a_1, \ldots, a_n \in_R \mathbb{Z}_q; \\ z_1 = g^{r_1}, \ldots, z_n = g^{r_n}, w = g^s; \\ x_1 = g^{a_1}, \ldots, x_n = g^{a_n}; \\ X = x_1 \cdots x_n; \\ Y_2 = X \cdot x_2^{-1}, \ldots, Y_n = X \cdot x_n^{-1} \end{array} \right\},$$

where $\mathsf{T} = (w, z_2, \ldots, z_n, Y_2, \ldots, Y_n)$ and $K = H(Y_2, \ldots, Y_n, X)$.

**Lemma 1.** *Let $\mathcal{D}$ be a distinguisher that, given $(\mathsf{T}, K)$ coming from one of the two distributions* Real *and* Fake*, runs in time $t$ and outputs 0 or 1. Then we have:*

$$\begin{aligned} &\big|\Pr[\mathcal{D}(\mathsf{T}, K) = 1 \mid (\mathsf{T}, K) \leftarrow \mathsf{Real}] - \\ &\qquad\qquad \Pr[\mathcal{D}(\mathsf{T}, K) = 1 \mid (\mathsf{T}, K) \leftarrow \mathsf{Fake}]\big| \\ &\qquad\qquad\qquad\qquad \leq \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t + (4n - 6)t_{exp}), \end{aligned}$$

*where $t_{exp}$ is the time required to compute an exponentiation in $\mathbb{G}$.*

*Proof.* We prove the lemma by using the random self-reducibility of the DDH problem. Consider the following distribution, which is constructed from the triple $(g^s, g^{r_2}, g^{s'r_2}) \in \mathbb{G}^3$:

$$\mathsf{Dist} = \left\{ (\mathsf{T}, K) \;\middle|\; \begin{array}{l} r_1, \alpha_3, \beta_3, \ldots, \alpha_n, \beta_n \in_R \mathbb{Z}_q; \\ z_1 = g^{r_1}, z_2 = g^{r_2}, \\ z_3 = g^{r_1\alpha_3 + r_2\beta_3}, \ldots, z_n = g^{r_1\alpha_n + r_2\beta_n}, w = g^s; \\ x_1 = g^{sr_1}, x_2 = g^{s'r_2}, \\ x_3 = g^{sr_1\alpha_3 + s'r_2\beta_3}, \ldots, x_n = g^{sr_1\alpha_n + s'r_2\beta_n}; \\ X = x_1 \cdots x_n; \\ Y_2 = X \cdot x_2^{-1}, \ldots, Y_n = X \cdot x_n^{-1} \end{array} \right\},$$

where $\mathsf{T}$ and $K$ are as defined above. If $(g^s, g^{r_2}, g^{s'r_2})$ is a Diffie-Hellman triple (i.e., $s = s'$), we have $\mathsf{Dist} \equiv \mathsf{Real}$ since $x_i = z_i^s$ for all $i \in [1, n]$. If instead $(g^s, g^{r_2}, g^{s'r_2})$ is a random triple, it is clear that $\mathsf{Dist} \equiv \mathsf{Fake}$.

**Lemma 2.** *For any (computationally unbounded) adversary $\mathcal{A}$, we have:*

$$\Pr[\mathcal{A}(\mathsf{T}, K_b) = b \mid (\mathsf{T}, K_1) \leftarrow \mathsf{Fake}; K_0 \leftarrow \{0,1\}^\ell; b \leftarrow \{0,1\}] = 1/2.$$

*Proof.* In experiment $\mathsf{Fake}$, the transcript $\mathsf{T}$ constrains the values $a_i$ by the following $n-1$ equations:

$$\begin{aligned} \log_g y_2 &= -a_2 + \textstyle\sum_{i=1}^n a_i, \\ &\vdots \\ \log_g y_n &= -a_n + \textstyle\sum_{i=1}^n a_i. \end{aligned}$$

Since $\mathsf{T}$ does not constrain the values $a_i$ any further and since the equation $\log_g X = \sum_{i=1}^n a_i$ is not expressible as a linear combination of the $n-1$ equations above, we have that the value of $X$ is independent of $\mathsf{T}$. This implies that

$$\Pr[\mathcal{A}(\mathsf{T}, X_b) = b \mid (\mathsf{T}, X_1) \leftarrow \mathsf{Fake}; X_0 \leftarrow \mathbb{G}; b \leftarrow \{0,1\}] = 1/2.$$

Then, since $H$ is a random oracle, the statement of Lemma 2 immediately follows.

**Theorem 1.** *Let $\mathcal{A}$ be a passive adversary attacking the protocol and running in time $t$. Then we have*

$$\Pr[\mathcal{A}(\mathsf{T}, K_b) = b \mid (\mathsf{T}, K_1) \leftarrow \mathsf{Real}; K_0 \leftarrow \{0,1\}^\ell; b \leftarrow \{0,1\}] \leq 1/2 + \mathsf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(t'),$$

*where $t' = t + O(nQt_{exp})$, with $Q$ being the number of protocol transcripts obtained by $\mathcal{A}$.*

*Proof.* This immediately follows from the lemmas 1 and 2 above, and the random self-reducibility of the DDH problem.

## 7 Conclusion

In this paper we have provided an efficient solution to the growing problem of contributory group key agreement over combined wired/wireless networks, which consist of both small mobile devices with limited computational resources and general-purpose computing machines with relatively high computing power. Our scheme takes only a constant number of communication rounds while achieving optimal message complexity. Furthermore, by spreading most of workload across the high power users, the scheme offers a low, fixed amount of computations to its mobile users and bounds the computational complexity of the other users by $O(\sqrt{n})$. A typical research topic is to improve the scheme to reduce the number of communication rounds while bounding the maximum complexity of any user by $O(\sqrt{n})$. We believe this is possible, and are currently working on it.

## References


1. G. Ateniese, M. Steiner, and G. Tsudik: New multiparty authentication services and key agreement protocols. IEEE Journal on Selected Areas in Communications, vol.18, no.4, pp. 628–639, April 2000.
2. N. Borisov, I. Goldberg, and D. Wagner: Intercepting mobile communications: The insecurity of 802.11. In Proc. of ACM MobiCom'01, pp. 180–189, 2001.
3. K. Becker and U. Wille: Communication complexity of group key distribution. In Proc. of ACM CCS'98, pp. 1–6, 1998.
4. C. Boyd and J.M.G. Nieto: Round-optimal contributory conference key agreement. In Proc. of PKC'03, LNCS 2567, pp. 161–174, 2003.
5. E. Bresson and D. Catalano: Constant round authenticated group key agreement via distributed computation. In Proc. of PKC'04, LNCS 2947, pp. 115–129, 2004.
6. E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval: Mutual authentication and group key agreement for low-power mobile devices. In Proc. of the 5th IFIP-TC6 International Conference on Mobile and Wireless Communications Networks (MWCN'03), pp. 59–62, 2003.
7. E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater: Provably authenticated group Diffie-Hellman key exchange. In Proc. of ACM CCS'01, pp. 255–264, 2001.
8. M. Burmester and Y. Desmedt: A secure and efficient conference key distribution system. Eurocrypt'94, LNCS 950, pp. 275–286, 1994.
9. W. Diffie and M.E. Hellman: New Directions in cryptography. IEEE Trans. on Information Theory, vol.22, pp. 644–654, 1976.
10. W. Diffie, P. Oorschot, and M. Wiener: Authentication and authenticated key exchanges. Designs, Codes, and Cryptography, vol.2, no.2, pp. 107–125, 1992.
11. I. Ingemarsson, D. Tang, and C. Wong: A conference key distribution system. IEEE Trans. on Information Theory, vol.28, no.5, pp. 714–720, September 1982.
12. J. Katz and M. Yung: Scalable protocols for authenticated group key exchange. Crypto'03, LNCS 2729, pp. 110–125, August 2003.
13. J. Nam, J. Lee, S. Kim, and D. Won: DDH-based group key agreement for mobile computing. Available at http://eprint.iacr.org/2004/127/.
14. A. Perrig, D. Song, and J.D. Tygar: ELK, a new protocol for efficient large-group key distribution. In Proc. of the IEEE Symposium on Security and Privacy, pp. 247–262, 2001.
15. M. Steiner, G. Tsudik, and M. Waidner: Diffie-Hellman key distribution extended to group communication, In Proc. of ACM CCS'96, pp. 31–37, 1996.
16. C. Wong, M. Gouda, and S. Lam: Secure group communications using key graphs. In Proc. of ACM SIGCOMM'98, pp. 68–79, 1998.