Diffie-Hellman Key Distribution in Wireless Multi-Way Relay Networks

Ronald Y. Chang, Sian-Jheng Lin, and Wei-Ho Chung
Research Center for Information Technology Innovation, Academia Sinica, Taiwan
Email: {rchang, sjlin, whc}@citi.sinica.edu.tw

***Abstract*— Diffie-Hellman protocol is a classical secret key exchange protocol for secure communications. This paper considers the extension of the original two-party Diffie-Hellman protocol to multi-party key distribution in wireless multi-way relay networks where multiple users can only communicate with one another through a single relay. Two efficient key exchange protocols are proposed. A performance comparison with existing methods adapted to the relay networks shows the enhanced efficiency and the originality of the proposed protocols designed specifically for the multi-way relay networks.**

***Index Terms*— Diffie-Hellman protocol, secret key distribution, multi-way relay networks.**

I. Introduction

Diffie-Hellman key exchange protocol, introduced in [1], is an efficient and elegant method that allows two users to establish a shared secret key over insecure channels. Its foundation is a simple mathematical fact: while computing the exponential over a finite field is easy, computing the logarithm (the inverse operation) over a finite field is difficult. In some cases with properly chosen prime number p, computing logarithms over the finite field $GF(p)$ is computationally impractical in a reasonable amount of time. Since its introduction, the elegancy of the Diffie-Hellman protocol and the growing applications that require secure communications have inspired many extensions to the original protocol. In [2] (and references therein), the Diffie-Hellman protocol has been extended to a group setting with more than two users.

Ever since the communication over the wireless medium has become prevalent, there has been significant interest in developing secret key exchange methods for secure wireless communications. The broadcast nature of the wireless medium facilitates the use of network coding (NC) [3] in the context of secret key distribution. In [4], the key identifiers (indices) to the maintained key database are exchanged through a relay node using NC. A related protocol is proposed in [5] for two-way relay networks, where secret keys are generated "on the fly" from radio propagation characteristics and exchanged using the idea of physical-layer network coding (PNC) [6]–[8]. The idea of utilizing radio propagation characteristics as the randomness source of secret keys has attracted significant attention recently (see, e.g., [9], [10]); however, it is still in the developing stage and its security is not as established and its use is not (yet) as widespread as the Diffie-Hellman protocol.

In this paper, we consider the extension of the two-party Diffie-Hellman protocol to n-party key distribution in wireless multi-way relay networks. Previous work on group Diffie-Hellman key distribution [2] does not consider the relay network and thus fails to exploit the new properties of the relay network. Previous work on key agreement for relay networks [4], [5] considers a different secrecy mechanism and does not directly apply to the Diffie-Hellman protocol. The current work presents two new Diffie-Hellman protocols to enhance the time efficiency of key distribution in multi-way relay networks, where one protocol has higher efficiency when n is small and the other has more amicable growth of complexity in large group key distribution scenarios.

This paper is organized as follows. Sec. II reviews the Diffie-Hellman protocol and its group extensions. The proposed general Diffie-Hellman protocols for multi-way relay networks are described in Sec. III and compared in Sec. IV. Conclusion is given in Sec. V.

II. Diffie-Hellman Key Distribution

A. Two-Party Diffie-Hellman Protocol

The Diffie-Hellman protocol was originally proposed in [1] for secret key exchange between two users named Alice and Bob who wish to communicate securely. The two users first agree on a large prime number p, which does not need to be kept secret. Let α be the primitive element of the finite field $GF(p)$. Then, Alice chooses a large random number a as her private key, computes $X_A = \alpha^a \bmod p$, and sends this value to Bob. Similarly, Bob chooses a large random number b as his private key, computes $X_B = \alpha^b \bmod p$, and sends this value to Alice. Alice and Bob then obtain their shared key $\alpha^{ab} \bmod p$ by raising their received signal to the power of their own private key over the finite field. Specifically, Alice computes

$$X_B^a \bmod p = (\alpha^b \bmod p)^a \bmod p = (\alpha^b)^a \bmod p \qquad (1)$$

and Bob computes

$$X_A^b \bmod p = (\alpha^a \bmod p)^b \bmod p = (\alpha^a)^b \bmod p. \qquad (2)$$

The security of the protocol is established upon the mathematical difficulty of computing logarithms over a finite field. Specifically, an eavesdropper observing X_A and X_B along with the known α and p cannot subsequently obtain a and b

This work was supported in part by the National Science Council, Taiwan, under grant NSC 102-2218-E-001-001.

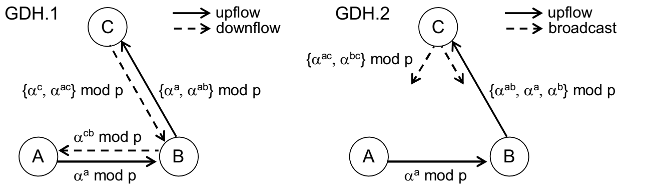

Fig. 1. Group Diffie-Hellman protocols GDH.1 and GDH.2 [2] for three users.

in a computationally feasible way for large p, and therefore cannot obtain the shared key between Alice and Bob.

B. Group Diffie-Hellman Protocols

The Diffie-Hellman protocol was extended to group settings in [2]. The proposed protocols enable each user to obtain the shared key which is in the form of the product of all users' private keys in the exponent. Here, we introduce the GDH.1 and GDH.2 protocols using the example of three parties, and refer the reader to [2] for the general protocol description. In the description here we drop $\bmod p$ for brevity, with the understanding that all operations are taken to be $\bmod p$. The GDH.1 protocol consists of an upflow stage and a downflow stage. The purpose of the upflow stage is to collect contributions from all users, and the purpose of the downflow stage is to forward useful intermediate values. As shown in Fig. 1, in the upflow stage, user A sends α^a to user B, and then user B sends α^a and α^{ab} to user C. In the downflow stage, user C sends α^c and α^{ac} to user B, and then user B sends α^{cb} to user A. Each user then constructs the shared key α^{abc} from the two-exponent value received. The GDH.2 protocol consists of an upflow stage and a broadcast stage. As shown in Fig. 1, in the upflow stage, user A sends α^a to user B, and then user B sends α^{ab}, α^a, and α^b to user C. In the broadcast stage, user C broadcasts α^{ac} and α^{bc} to all users. Similar to the original Diffie-Hellman protocol, an eavesdropper observing the intermediate values cannot use any combination of them to reproduce the shared key.

III. New Diffie-Hellman Protocols for Multi-Way Relay Networks

We consider a multi-way relay network with n ($n \geq 2$) users. In this network, direct communication between users is not available and users can only communicate through a single relay. The relay and the users operate in the half-duplex mode. We introduce two general Diffie-Hellman protocols for the multi-way relay network.

A. Protocol 1

Before describing the general mechanism of this protocol, we first look at some examples (special cases).

Example 1A: Consider $n = 2$. First, user A sends $X_A = \alpha^a \bmod p$ and user B sends $X_B = \alpha^b \bmod p$ to the relay concurrently. After receiving the signal sum, the relay performs a modulo-p operation and broadcasts

$$X_R = \left(\alpha^a \bmod p + \alpha^b \bmod p\right) \bmod p = (\alpha^a + \alpha^b) \bmod p$$

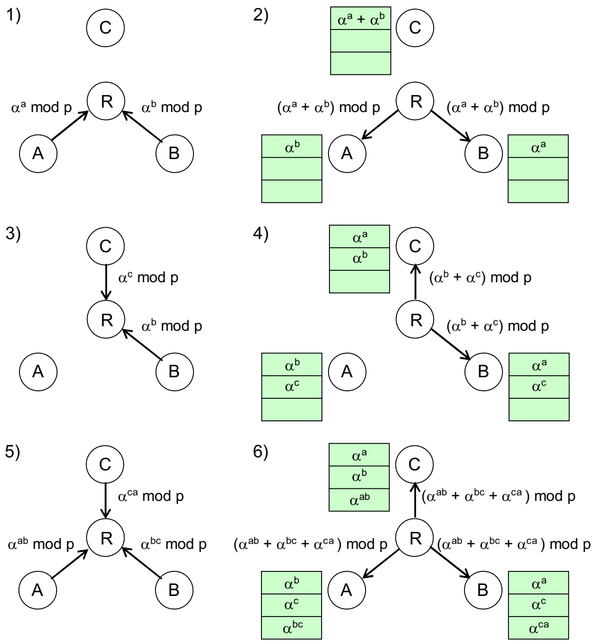

Fig. 2. Diffie-Hellman key distribution protocol 1 for three-way relay networks. The useful values obtained at each user after relay broadcasting are shown next to each user over colored background (with $\bmod p$ dropped for simplicity). Six time slots are needed.

to the two users. User A (or B) can use X_R and its own X_A (or X_B) to obtain X_B (or X_A). Then, similar to (1) and (2), users A and B can compute the shared key.

Example 1B: Consider $n = 3$. Fig. 2 summarizes the protocol operation. The protocol first initiates a two-user key exchange similar to the $n = 2$ case. In the first and second time slots, communication takes place between users A and B. After relay broadcasting in the second time slot, user A obtains α^b, user B obtains α^a, and user C obtains $\alpha^a + \alpha^b$. In the third and fourth time slots, communication takes place between users B and C. After relay broadcasting in the fourth time slot, user A obtains α^c through the knowledge of α^b, user B obtains α^c, and user C obtains α^b, which, along with the knowledge of $\alpha^a + \alpha^b$, allows user C to obtain α^a. Thus, after four time slots, each user has obtained the one-exponent values about all the other users. In the fifth time slot, each user raises one of the obtained one-exponent values to the power of its own private key and sends the two-exponent value to the relay concurrently. Specifically, user A sends α^{ab}, user B sends α^{bc}, and user C sends α^{ca}. Upon receiving the data, the relay performs a modulo-p operation and broadcasts $\alpha^{ab} + \alpha^{bc} + \alpha^{ca}$ in the sixth time slot. Now, since user A knows α^{ab} and α^{ca} already (by taking α^b and α^c to the power of a), user A can subtract them from the received value to obtain α^{bc}. Similarly, user B can obtain α^{ca} and user C can obtain α^{ab}. From these two-exponent values each user can easily compute the shared key α^{abc}.

General protocol: To generalize the protocol to n users, note that each user needs to obtain the ($n - 1$)-exponent value with its own exponent excluded. The proposed protocol initiates the communication of one-exponent values in the first phase, followed by the communication of two-exponent values in the second phase, followed by the communication of three-exponent values in the third phase, etc., until the communication of $(n-1)$-exponent values in the $(n-1)$th phase. For each user, there are $\binom{n-1}{k}$ k-exponent values that do not involve its own exponent. This protocol enables each user to obtain all these $\binom{n-1}{k}$ k-exponent values after the kth phase consisting of $2\binom{n-1}{k}$ time slots, where the factor of 2 reflects one transmission from the users to the relay and one transmission from the relay to the users. In the kth phase, every communication from the users to the relay involves exactly $k+1$ users jointly transmitting to the relay, i.e., a $(k+1)$-way transmission. Let $\mathcal{U} = \{A, B, C, \ldots\}$ be the set of users and $\mathcal{N} = \{a, b, c, \ldots\}$ be the set of their private keys (exponents). The protocol is summarized as follows.

```
Protocol 1: $k$th phase $(k = 1, \ldots, n-1)$
  Arbitrarily select $\binom{n-1}{k}$ distinct $(k+1)$-element subsets
  of $\mathcal{U}$.[1] Let $\mathcal{U}^j_{k+1}$ be the subset and $\mathcal{N}^j_{k+1}$ be the
  corresponding set of keys, $j = 1, \ldots, \binom{n-1}{k}$.
  for $j = 1$ to $\binom{n-1}{k}$ do
      1. Each user $U \in \mathcal{U}^j_{k+1}$ sends the product of $k$
      cyclically consecutive elements in $\mathcal{N}^j_{k+1}$ starting at
      element $u$ in the exponent, in the joint $(k+1)$-way
      transmission to the relay.
      2. The relay broadcasts the received sum value (after
      taking $\bmod p$) to the users.
```

In the kth phase, for each user (say, user A) involved in one $(k+1)$-way transmission to the relay there is exactly one transmitted k-exponent value that does not involve user A's private key a and therefore can be decoded at user A by subtracting from the signal broadcasted from the relay all other k-exponent values that involve a (e.g., in Example 1B, in the second phase user A sends α^{ab}, user B sends α^{bc}, and user C sends α^{ca}, and there is only one two-exponent value that does not involve a and can be decoded at user A). The k-exponent values that involve a to be subtracted from the broadcasted signal can be computed at user A by simply taking the $(k-1)$-exponent values with its own exponent excluded (known from the previous phase) to the power of a. After $\binom{n-1}{k}$ times of $(k+1)$-way communication in the kth phase, each user obtains all the $\binom{n-1}{k}$ k-exponent values that do not involve its own exponent.

Property 1: The general Protocol 1 requires $2^n - 2$ time slots for key distribution.

This follows from the fact that the kth phase requires $2\binom{n-1}{k}$ time slots and there are $n-1$ phases; thus, the total number of time slots is given by $\sum_{k=1}^{n-1} 2\binom{n-1}{k} = 2^n - 2$.

B. Protocol 2

A naive protocol to construct the shared key can make each user collect the contribution of all other $n-1$ users sequentially in $2(n-1)$ time slots and repeat the same procedure for all users, resulting in $2n(n-1)$ time slots in total. In this scheme, a polynomial time with respect to n is required as opposed to an exponential time in Protocol 1. In the proposed Protocol 2 we aim to produce an improved polynomial-time protocol. Protocol 2 performs multiple phases of communication similar to Protocol 1. In Protocol 2, each phase requires a fixed number of $2(n-1)$ time slots, and in any phase each communication from the users to the relay involves exactly two users jointly transmitting to the relay, i.e., a two-way transmission. The set of jointly transmitting users is given by the two-consecutive-element subset of $\mathcal{U}$ (e.g., for $\mathcal{U} = \{A, B, C, D\}$, the two-consecutive-element subsets are $\{A, B\}$, $\{B, C\}$, and $\{C, D\}$). The protocol is summarized as follows.

```
Protocol 2: $k$th phase $(k = 1, \ldots, n-1)$
  Let $\mathcal{U}^j_{2c}$ be the two-consecutive-element subset of $\mathcal{U}$,
  $j = 1, \ldots, n-1$.
  for $j = 1$ to $n-1$ do
      1. Each user $U \in \mathcal{U}^j_{2c}$ sends the product of $k$
      cyclically consecutive elements in $\mathcal{N}$ ending at
      element $u$ in the exponent, in the joint two-way
      transmission to the relay.
      2. The relay broadcasts the received sum value (after
      taking $\bmod p$) to the users.
```

In the kth phase, the rotated two-way transmissions allow each user to obtain all the $n-1$ k-exponent values transmitted by other users in this phase due to the chain relation of two-way transmissions. Among the $n-1$ k-exponent values there is one which is the product of each user's own key's cyclically previous k consecutive elements in $\mathcal{N}$ (e.g., for $\mathcal{N} = \{a, b, c, d\}$, the product of the cyclically previous k consecutive elements in $\mathcal{N}$ for user A is d for $k = 1$, cd for $k = 2$, and bcd for $k = 3$). Thus, after the $(n-1)$th phase, each user will obtain the desired $(n-1)$-exponent value that does not involve its own exponent. An example of this protocol is shown in Fig. 3 for $n = 3$.

Property 2: The general Protocol 2 requires $2(n-1)^2$ time slots for key distribution.

IV. Comparison

A. Comparison with the Protocols in [2]

Here, we compare the proposed protocols with the group Diffie-Hellman protocols GDH.1 and GDH.2 proposed in [2]. To be able to do this, we need to first adapt these protocols, developed for the scenario where users can communicate directly with each other, to the relay setting. This task is in general nontrivial. The reason is that the broadcast nature of the relay may allow unintended users of a transmission to

[1] Note that this is always possible since $\binom{n}{k+1} = \binom{n-1}{k} + \binom{n-1}{k+1}$ for all integers n, $k \geq 0$.

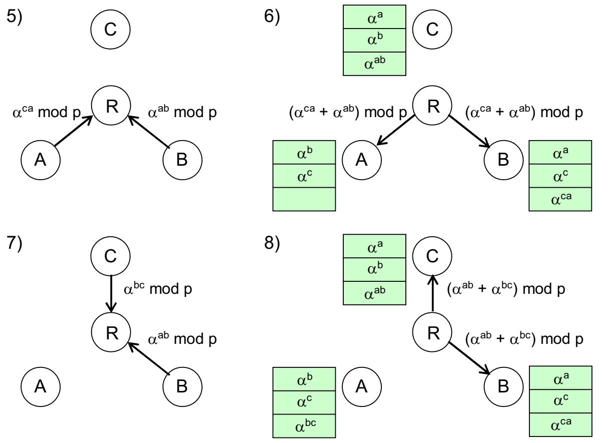

Fig. 3. Diffie-Hellman key distribution protocol 2 for three-way relay networks. The first four time slots (not shown) are identical to the first four time slots in Fig. 2. Eight time slots are needed.

also receive the message earlier than its scheduled time in the original protocol and thus renders the scheduled transmission unnecessary. In addition, while multiple values can be transmitted from one user to another in one batch in GDH.1 and GDH.2, only one value can be transmitted at a time in our consideration. In light of the difficulty of an adaptation for the general case, we consider an adaptation for the special cases of $n = 2$ and $n = 3$, and compare the adapted protocols with our methods.

For $n = 2$, it is easy to see that GDH.1 and GDH.2 require four time slots in the two-way relay network. The proposed Protocol 1 and Protocol 2 require two time slots, achieving the minimum number of time slots for each user to contribute (transmit) its own exponent and receive the other user's exponent in half-duplex mode.

For $n = 3$, the GDH.1 protocol described in Sec. II-B has five distinct values in communication. Adapted to the relay setting, GDH.1 requires 10 time slots (note that the sending of α^a from user B to user C in the upflow stage can be waived since user C has received it earlier when the relay broadcasts α^a for user A). The GDH.2 protocol operates differently but can be analyzed similarly to yield 10 time slots. In comparison, the proposed Protocol 1 and Protocol 2 require six and eight time slots, respectively.

B. Comparison between Protocol 1 and Protocol 2

With Protocol 2, in general, each user will not obtain all the $\binom{n-1}{k}$ k-exponent values after the kth phase as in Protocol 1, but only those that will allow each user to help its cyclically next user in $\mathcal{U}$ to obtain the desired $(n-1)$-exponent value. This leads to some loss of efficiency when n is small, but avoids the exponential growth in the time slots required. In Fig. 4, we plot the number of time slots required versus the number of users in the network for the proposed two protocols. As can be seen, Protocol 1 is more efficient for key distribution among $n \leq 5$ users, but does not scale well with the increase of n. Thus, Protocol 1 is of interest mainly when n is small, which is the typical application for key distribution. For large group key distribution, Protocol 2 is more efficient and its advantages increase as n increases.

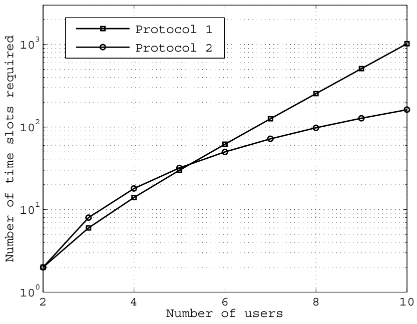

Fig. 4. The number of time slots required for key distribution versus the number of users, for Protocol 1 and Protocol 2.

V. Conclusion

We have developed two general Diffie-Hellman key distribution protocols for the multi-way relay network, one with high efficiency in small group settings and the other with practical efficiency in large group settings. A comparison with previous methods adapted to the same setting suggested that the proposed protocols are not derivative variations of the existing ones. Future works include a study of the theoretical limit of Diffie-Hellman key distribution in the multi-way relay network. In addition, as we conjecture that Protocol 1 is optimal for $n = 3$ in terms of certain aspects of protocol efficiency, it remains to perform a rigorous study of the proposed protocols against the theoretical optimum.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[2] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *Proc. ACM Computer and Communications Security Conference (CCS)*, New Delhi, India, Mar. 1996, pp. 31–37.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[4] P. F. Oliveira and J. Barros, "A network coding approach to secret key distribution," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 414–423, Sept. 2008.

[5] T. Shimizu, H. Iwai, and H. Sasaoka, "Physical-layer secret key agreement in two-way wireless relaying systems," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 650–660, Sept. 2011.

[6] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. ACM MobiCom*, Los Angeles, CA, Sept. 2006, pp. 358–365.

[7] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Proc. IEEE ICC*, Istanbul, Turkey, June 2006, pp. 3885–3890.

[8] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proc. IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.

[9] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proc. ACM MobiCom*, San Francisco, CA, Sept. 2008, pp. 128–139.

[10] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. ACM MobiCom*, Beijing, China, Sept. 2009, pp. 321–332.