A Family of Protocols for Group Key Generation in Ad Hoc Networks

Alec Yasinsac Vikram Thakur Stephen Carter Ilkay Cubukcu

Computer Science Department
Florida State University
yasinsac@cs.fsu.edu

Abstract

With the pervasive distribution of highly mobile computing devices, establishing dynamic networks among these mobile nodes is a growing demand. This new field of study has come to be known as ad hoc networking. Because of the nature of ad hoc networks, protecting communication in this environment is difficult, with solutions based on cryptographic techniques. Most existing group key management techniques are not suited to the ad hoc network environment.

In this paper we give a family of efficient cryptographic protocols for establishing secure groups in the ad hoc network environment. We begin by detailing the foundational protocol based on the Diffie-Hellman key exchange and show how this protocol is efficient and secure. We go on to give protocols for group join and exclusion, and a corresponding set of authenticating group protocols based on our foundational protocol.

Keywords: Cryptographic Protocols, Ad Hoc Networking, Security, Conference Key Establishment, Group Keys

1. Introduction

Ad hoc networks are a hot research topic. The enabling technology for this field includes: (1) Reduction in size of chips and the nodes that hold them (2) Dramatic improvements in wireless communication speed, bandwidth, and reliability. Because they can, people want to move about freely, with their computers turned on and connected. Ad hoc networks are a natural result of user demand meeting the enabling technology.

Highly mobile devices that dynamically organize ad hoc networks, intercommunicate, pass information to other wireless users, and then dissolve characterize ad hoc networks. An essential characteristic of ad hoc networks is the ability to dynamically form communications groups. The inherently chaotic nature of these groups complicates protecting communications security for these groups.

There are many proposed cryptographic solutions for group communication in the literature [STW00, STW96, AST00, AST98, FW93, BD95, HK89]. Unfortunately, most of these protocols either require structure that is neither desired nor available in ad hoc networks, or are resource intensive. In this paper, we derive a simple, efficient family of protocols specifically to support ad hoc networks.

2. Ad Hoc Network Group Key Establishment

We first outline the environment that we consider for ad hoc group establishment. Our vision is a set of communicating nodes characterized by highly dynamic membership, with short membership duration and a large number of joins and drops. Such dynamic group membership is illustrated in Figure 1.

The primary communication medium is wireless broadcast, where most communicating parties receive each message. Messages are relayed [routed] by some nodes, but not necessarily by all. These networks may be densely populated, where nodes receive a high volume of, sometimes rebroadcast, messages. They may also be sparse, where most communication is comprised of relayed point-to-point messages, and where single links may connect sub-groups, so loss of a single link can separate large subgroups.

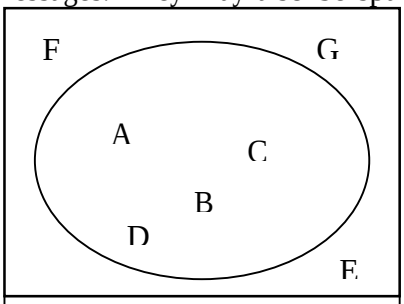

Figure 1a
Time = T_1

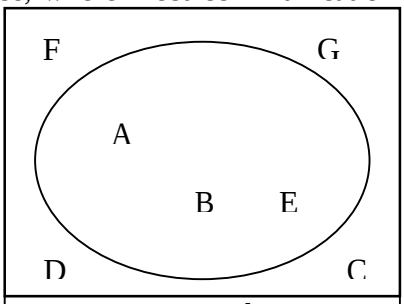

Figure 1b
Time = T_2

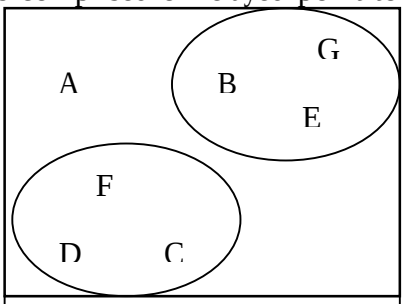

Figure 1c
Time = T_3

The communication medium is not as important here as is the flavor of the environment that we envision. Ad hoc networks may be connected by any number of heterogeneous communications medium, including radio, infrared, laser, and even dynamic wire and fiber-optic links. We go so far as to point out that fixed communications sites may join in the ad hoc networks, but draw the line where any dependence is given to such existing infrastructure.

Specifically, we consider networks as being ad hoc if they have no required, permanent infrastructure. While towers (as those that support cellular networks) are not required, we do not exclude their presence from the environment. Still, we consider that most nodes have short range, low power transmission capability. Our emphasis is on the "ad hocness" of the network. Members come and go at varying paces and with varying throughput requirements and capabilities.

The environment that we describe demands efficient protocols, that limit both the number and size of messages and in the number of computations required in each round. As we noted earlier, there have been a number of different group management and group key established protocols proposed in the literature. The most widely published protocol structure is that proposed by Steiner, et al [STW00]. We now give a quick overview of the CLIQUES approach to group key establishment.

2.1. Overview of CLIQUES

CLIQUES is a family of protocols for contributory and authenticated group key distribution, based on the Diffie-Hellman (DH) key exchange process [DH76]. In the most simple of the CLIQUES protocols (IKA.2 [STW00]), the key computation proceeds from node to node, with each node raising the previous computations to the power of their private DH value. The final node in the computation string generates the values that each previous node needs in order to compute the final (group) key and transmits all these values in a broadcast message.

While innovative in their approach, there are a number of characteristics of these protocols that limit their utility in ad hoc networks. First, the station-to-station nature of the suite necessitates serial execution of the computations. For a large number of nodes in a highly dynamic environment, this is a critical inefficiency. Additionally, in order to execute a serial computation, the nodes must be serialized. Most critically, the final node in the computation must recognize their position. Such architectural limitations prove both tricky and restrictive, properties that do not fit well in the dynamic, ad hoc environment

2.2. An Optimal Ad Hoc Group Key Agreement Protocol

The CLIQUE family of protocols is based on the DH computation with the number of messages and computations on the order of n, the number of nodes. We now offer a foundational protocol, also based on the DH computation, that avoids much of the restrictive nature of the CLIQUES protocols. Most essentially, (1) There is no requirement for serialization and (2) The number of messages required is optimal.

2.2.1 The Foundational Protocol

In addition to its efficiency, our protocol family is simple. The fundamental protocol consists only of message two rounds[1]. In round one, each member broadcasts its public DH number. In the second round, a group coordinator broadcasts sufficient information that each member can compute the fully contributory group key. More formally, the foundational protocol proceeds as follows:

1. One member announces formation of a group
2. The potential group members (i = 1…n) select[2] and publish a coordinator (member #0), the DH base g and modulus p[3] with the base and modulus having all the necessary properties to ensure that the impending DH computations are secure.
3. Each i^{th} member (except the coordinator) chooses a random x_i as their private DH number and broadcasts their public DH number g^{r_i}
4. The coordinator generates the random numbers z and x_0, g^{r_0}, $g^{r_i x_0}$ for each i, and encrypts[4] $e[z]g^{r_i x_0}$ for each i. The coordinator then concatenates g^{r_0} with all the encrypted values and broadcasts the concatenated message.
5. After receiving the broadcast from the coordinator, each member computes $g^{r_i x_0}$ using their private x_i, decrypts z, and computes a combining function $F = f(g^{r_1}, g^{r_2} \quad g^{r_i})$, and the group key, $K = g^{F \circ z}$.

The functions f and ◦ ensure member contribution and the security of the group key. We discuss desirable properties and offer some suggestions for these functions in Section 3.

3. Positive Properties of the Scheme

There are numerous positive properties of this protocol. First, it is simple. The intent of the step is clear and the computations have proven to be secure over years of use as the two-party DH protocol.

---

[1] We define a round as a group of messages all having the same purpose, that all must be completed before the next step (or round) may be accomplished.

[2] Coordinator election may or may not require message transmissions and may be as simple as members taking turns, or using the "I called it" paradigm. More sophisticated election mechanisms may be in order in high-risk environments.

[3] All computations throughout the paper (encryption excepted) are mod p.

[4] Using the same encryption algorithm that protects group communications

Secondly, the protocol produces a key that is verifiably contributory. When each member computes the key, its own contributory component is incorporated along with the components provided by the other members.

Moreover, as we noted earlier, our protocol is highly efficient. It is optimal in the number of messages required for an environment that is free of infrastructure, such as prior shared secrets (one message per node) and in the total amount of data transmitted (one value the size of the modulus per node). There are minimal computations and setup required, yet the protocol is as strong as the weaker of the encryption algorithm of choice and the DH computation.

Additionally, the protocol is fully distributed. Though a coordinator is elected in each run, the coordinator position is not [necessarily] persistent and does not represent a bottleneck or single point of failure. If the coordinator fails, another coordinator is summarily elected and the protocol proceeds.

Finally, our fundamental protocol can be easily modified for key verification, authenticated key exchange, and group join and exclusion operations, all while retaining its positive properties for security, efficiency, and robustness. In the following sections, we give a set of protocols for these functions. First, we address similarities between another existing conference key establishment scheme.

3.1.1 Similarities to Burmester and Desmedt

In [BD95] Burmester and Desmedt proposed several protocols for conference key distribution and offered modifications for efficiency and authentication in [BD96]. There are a number of similarities and distinguishing characteristics between our scheme and that offered by Burmester and Desmedt. In this section, we focus on the protocols offered in Sections 3.1 and 3.4 of [BD95], which we will respectively term "star" and "neighbors" protocols.

Though not clearly depicted as a broadcast protocol in the paper, the star protocol is similar in structure to our scheme. The group members transmit their DH public value and the group coordinator responds with the group key, disguised by multiplication with the DH key that they share. A distinguishing characteristic of our protocol here is that the key that they distribute is not contributory. We believe that it is essential that group keys utilized in highly dynamic, ad hoc groups be contributory. Burmester and Desmedt seem to acknowledge this principal by designating the neighbors protocol, which produces a contributory group key, as their main protocol.

The neighbors protocol generates a contributory key in two rounds by having each node generate a very novel, DH-type computation combining their private DH value with the public DH value of each of two immediate neighbors. While slightly more complex than the star protocol, each member is able to generate the fully contributory group key from the two rounds of broadcasts.

While this solves our earlier concern regarding the contributory nature of the resulting group key, it introduces another pitfall in ad hoc communications: structure of the group is required. We accept the necessity for some reasonable setup for conference key distribution, and acknowledge that our scheme requires some mechanism for electing a group coordinator, though no other structure is required for our scheme.

On the other hand, the structure for the neighbors protocol goes well beyond coordinator selection, effectively requiring a group serialization that must be agreed upon by each member before a group key can be established. For small groups, such an algorithm may prove unwieldy; for large groups, it would be resource intensive and problematic. We posit that mechanisms requiring less structure are better suited to ad hoc groups than are schemes that require more structure.

3.2. Efficient Protocols for Group Join and Exclusion

Single member join and exclusion operations under our scheme are extremely simple and require only two message transmissions and n computations. As with the foundational protocol, these protocols are fully contributory and ensure key independence. We assume the following groups were established with our foundational protocol and that each member retained the public DH values of the other group members.

3.2.1 Single Member Group Join

Since each group member already has the public DH values of all group members, all that is needed is for those values to be given to the new member, the new members public DH value to be given to the existing members, and a new seed to be given to both the existing and new group members. The protocol proceeds as follows:

1. A new potential member generates their private DH value x_i, and announces their desire to join the group, along with their public DH number, g^{x_i}.
2. If a new group coordinator is required, or desired, one is elected from among existing members.
3. The group coordinator broadcasts the following in a single message:
   a. The DH base g and modulus p
   b. The public DH numbers for the existing group members
   c. The new seed z' individually encrypted as $e[z']g^{r_i x_0}$ for each i (including the new member).
4. After receiving the broadcast from the coordinator, each member computes $g^{r_i x_0}$ using their private x_i, decrypts z', and computes $F = f(g^{r_1}, g^{r_2} \quad g^{r_i})$, and the group key, $K = g^{F \circ z'}$.

3.2.2 A Single Member Exclusion Protocol

Excluding a single member from the group is even simpler than the join operation shown in the previous section. All that is needed is to generate and distribute a new seed to the existing group. Again, we assume that each group member retains the publicly broadcast DH numbers of all group members. The protocol follows:

1. A member announces their intent to leave the group.
2. If a new group coordinator is required, or desired, one is elected
3. The group coordinator broadcasts the new seed z', individually encrypted as $e[z']g^{r_i x_0}$ for each remaining i^{th} member, and concatenated into a single message.
4. After receiving the broadcast from the coordinator, each member computes $g^{r_i x_0}$ using their private x_i, decrypts z', and computes $F = f(g^{r_1}, g^{r_2} \quad g^{r_i})$, and the group key, $K = g^{F \circ z'}$.

3.3. Authenticated Key Exchange

Like the original DH two party key exchange, the protocols that we have proposed to this point provide only secure, contributory key exchange. They do not ensure member authentication, leaving open the question of exactly who the members are after the group key is established.

Unfortunately, authentication [i.e. verifying the identity of the parties] is fundamentally harder than secure key exchange. Much has been written about the nature of authentication [LABW91, RS97, Gol96]. It is generally accepted that authentication requires either prior knowledge between the prover and verifier to firmly establish identity, or a mutual relationship with a third party that can verify identity. This assumption is commonly modeled as an assumed Public Key Infrastructure.

We note here that assuming that an effective Public Key Infrastructure (PKI) is in place is dangerous. The idea has been around for some time, but establishing an effective PKI is proving to be a significant challenge. Some have even suggested that is neither possible nor desirable to establish such an infrastructure [ES00]. Nonetheless, we show how our key exchange scheme can produce authenticated group key exchange suitable for ad hoc networks, if such an infrastructure is in place.

Specifically, for the following protocol, we assume that a system of identity certificates is in place. We do not suggest whether these certificates are verified by a trusted third party or through other public key type infrastructure; only that certificates can be accurately verified. We further assume complete certificate distribution[5] (i.e. all members and potential group members have free access to all member and potential member certificates).

Finally, the structure of the public keys and their corresponding private keys are the traditional DH key pair, generated under a well-known base g and modulus p. With these assumptions in place, the one message, authenticated, ad hoc, group key exchange protocol proceeds as follows:

1. An originating member decides to form a group. This originating member becomes the group coordinator[6] (member #0). The coordinator generates the random number z, forms the array G containing the identities of the group members, computes $g^{r_i x_0}$ for each i, and encrypts $e[z,G]g^{r_i x_0}$ for each i. The coordinator then concatenates the encrypted values and broadcasts the concatenated message.
2. After receiving the broadcast from the coordinator, each member computes $g^{r_i x_0}$ using their private value x_i, decrypts the value z, and computes both $F = f(g^{r_1}, g^{r_2} \quad g^{r_i})$, and the group key, $K = g^{F \circ z}$.

This protocol is optimal in the number of messages (one) and efficient in the size of transmitted information and its number of computations. All members are authenticated to each other, and group membership authenticity is as strong as the certificate verification system.

4. A Few Words About Function Selection in the Foundational Protocol

The security of the family of ad hoc, group key management protocols that we propose turns on our ability to generate a suitable exponent for g to produce the final key. This exponent must have three primary properties:

1. It must be formed from the public keys of the group members
2. It must be random so that it does not compromise the security of the final key, and
3. It must be deterministic so that it can be computed by each member of the group.

One option for function selection is to let $\circ$ be multiplication and f be a secure hash function. Then, since we require that z be random, the final key K is as strong as the classic two-party DH key. Secure hash functions are abundant, well understood, and efficient to compute.

Another option is to select the XOR operation as $\circ$. This relaxes the requirements on f so f may simply be selected to be multiplication mod p.

In an environment where group membership is highly dynamic, key lifetime is short, and little data is available for cryptanalysis, it is unlikely that the strength of f will be a vulnerability under this scheme. Still, there are potential pitfalls to watch out for in selecting f. For example, one should

---

[5] Again, this is a dangerous assumption. Effective distribution/acquisition of valid certificates is not a solved problem. Still, it is a standard assumption made to allow research to continue on authenticated services.

[6] Alternatively, if desired, the originating member can announce the desired group and a coordinator may be elected.

avoid selecting any function as f that has more than a trivially high probably of evaluating to zero (0) or one (1).

5. Conclusion

In this paper, we define an environment for efficient, secure, ad hoc group key exchange. We describe why existing group key exchange protocols and show how two widely publicized group protocol approaches do not meet the needs of the ad hoc environment because of efficiency, architectural considerations, and setup requirements.

Our main contribution is to give a family of efficient protocols for the ad hoc network environment. We begin by detailing the foundational protocol based on the Diffie-Hellman key exchange and show how this protocol is efficient and secure. We then give protocols for group join and exclusion, and a corresponding set of authenticating group protocols based on our foundational protocol.

6. Bibliography

[AST00] Guisseppe Ateniese, Michael Steiner, Gene Tsudik, "New Mulitparty uthentication Services and Key Agreement Protocols", IEEE Journal of Selected Areas in Communications, Vol. 18, No. 4, (Apr 2000): pp 1-13

[AST98] Guisseppe Ateniese, Michael Steiner, Gene Tsudik, "Authenticated Group Key Agreement and Friends", In ACM CCS ACM, November 1998

[BD95] M. V. D. Burmester and Y. Desmedt. "A Secure and Efficient Conference Key Distribution System", In A. D. Santis, editor, Advances in Cryptology -- EUROCRYPT '94, volume 950 of Lecture Notes in Computer Science, pages 275--286. Springer-Verlag, 1995.

[BD96] Mike Burmester and Yvo Desmedt, "Efficient and secure conference-key distribution", Proc. 1996 Security Protocols Workshop, Cambridge, UK, Apr. 10-13, 1996, Springer-Verlag LNCS 1189, pp. 119-129

[CD85] D.R. Cheriton and S.E. Deering. “Host Groups: A Multicast Extension for Datagram Internetworks”. In 9th Data Communication Symposium, September 1985

[CGIMNP99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. “Multicast security: A taxonomy and some efficient constructions”, In *Proc. IEEE Infocom*, March 1999

[DOW92] W. Diffie, P. C. van Oorshot, and M. J. Wiener, “Authentication and Authenticated Key Exchanges”, *Designs, Codes and Cryptography*, 2(2):107-125, June 1992

[ES00] Carl Ellison and Bruce Schneier, "Ten Risks of PKI, What You Are Not Being Told About PKI", Computer Security Journal, Vol. XVI, No. 1, 2000

[FW93] Michael Fischer and Rebecca N. Wright, "An Efficient Protocol for Unconditionally Secure Secret Key Exchange", Proc of the 4th Symp on Discrete Algorithms (1993), pp. 475-83

[HK89] Lein Harn & Thomas Kiesler, "Authenticated Group Key Distribution Scheme For a Large Distributed Network", IEEE Symposium on Security and Privacy, 1989, pp 300-309

[GOL96] Dieter Gollmann. What do we mean by entity authentication? In Proceedings of the IEEE Symposium on Security and Privacy, pages 46-54. IEEE CS Press, May 1996.

[LABW91] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, 'Authentication in Distributed Systems: Theory and Practice', ASM OS Review, Vol 25, No. 5, Special Issue, Proceedings of the 13th Symposium on Operating System Principles, 13-16 Oct 1991, pp. 165-182

[Mea91] Meadows, C., 'A System for the Specification and Analysis of Key Management Protocols'. From 1991 IEEE Computer Society Symp on Research in Security and Privacy, pp. 182-195.

[MRR99] M. J. Moyer, J. R. Rao, and P. Rohatgi, “A survey of security issues in multicast communications”. *IEEE Network*, pp. 12-23, Nov/Dec 1999

[Rei94] M. Reiter, “A secure group membership protocol”, Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1994

[RS97] K. Reiter and S. Stubblebine, "Toward Acceptable Metrics of Authentication" Proceedings of the IEEE Symposium on Security and Privacy. pp10-20. May 1997

[STW00] Michael Steiner, Gene Tsudik, and Michael Waidner, Key Agreement in Dynamic Peer Groups", IEEE Transactions on Parallel and Distributed Systems", Vol. 1, No. 8 (Aug 2000): pp 769-80

[STW96] M. Steiner, G. Tsudik, M. Waidner, “Diffie-Hellman key distribution extended to group communication", In Proc. 3rd ACM CCS, New Dehli, India, 14-16 May, 1996, pp. 31-7

[WGL00] C. K. Wong, M. Gouda, and S. S. Lam, “Secure group communications using key graphs”, *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, February 2000