Journal of Computer Science 6 (3): 229-234, 2010
ISSN 1549-3636

A Framework for Group Key Management Protocol Assessment Independent of View Synchrony

David Manz, Paul Oman and Jim Alves-Foss
Department of Computer Science, University of Idaho, P.O. Box 441008, Moscow,
ID 83844-1008 USA

Abstract: Problem statement: As group key management extended into the area of large dynamic networks, complex issues emerged involving the many operations that run over several network topologies. The issues that occurred due to multiple topologies were also compounded by differing views of the network, taken at different time slices or positions within the network. This was especially complex when figuring in mobile, ad-hoc networks. View synchrony is the current operational technique, or assumption, applied to group key exchange protocols. However, before this analysis view synchrony was just that, an assumption and the literature for group key exchange lacked an inquiry into what could happen when view synchrony was removed. Current group key management protocols rely on view synchrony and yet all protocols vary in requisite operational descriptions and performance measures. In this study, a framework for group key management protocol operations and performance measures was defined and examined how that framework could be used to compare and contrast existing protocols with and, more importantly, without view synchrony. **Approach:** Current literature lacked categories by which to quantify the performance metric of the protocols. This study first defined the dynamic key operations that all protocols share. By these definitions, group key management protocols were directly compared. Once definitions existed, this study assembled a list of costs that every protocol requires to establish and share keys across the dynamic group. These results provided an understanding of view synchrony's role and whether or not it should be solely relied on in these current protocols. **Results:** The prior conclusion that view synchrony was an integral part of all group key management protocols was shattered, when seen through the lens of communication costs and assumptions in wireless ad-hoc networks. View synchrony, as an assumed part of all group key management was previously inconsistently portrayed. The ability to see this before did not exist because a framework upon which to evaluate the costs did not exist. Now, literature can proceed with clearly defined understandings of what values exist in group key management protocols. **Conclusion/Recommendations:** Better communication in group key management will be a benefit to the entire field. Now that costs can be analyzed, procedure and security can be improved and protocols can be implemented for wireless ad-hoc networks. In addition, it led two authors of this study to create a new protocol, DTEGK, to maximize the most efficient communication, as view synchrony was hindering the effectiveness of previous protocols. Without the hindrance of view synchrony and a quantitative list of defined communication costs, protocols can also now be extended into the wireless, ad-hoc realm of group key management.

Key words: Group key management, cryptography, wireless, ad-hoc networks

INTRODUCTION

In 1976, Diffie and Hellman (1976) introduced an implementation of two-party key exchange protocol that allowed two participants to create a private cryptographic key through the use of publicly exchanged messages.

Today, the Diffie-Hellman (DH) key exchange protocol and its many variants are commonly used for everyday, two-party secure messaging. Moreover, the concept of a cryptographic key exchange protocol has been extended to group key exchange protocols analogous to the DH protocol, but for dynamic groups of participants. Mobile Ad-hoc Networks (MANETs),

Corresponding Author: David Manz, Department of Computer Science, University of Idaho, P.O. Box 441008, Moscow, ID 83844-1008 USA Tel: 1-509-432-5492 Fax: 1-208-885-6840

like fleets of unmanned vehicles or sensor arrays, can make extensive use of group key exchanges as individual nodes on the network are added or removed from the swarm of communication defining the ad-hoc network (Manz *et al*., 2007). Wired and wireless emergency communications and disparate groups of military units operating in hostile theaters may also require dynamic group key exchange (Manz *et al*., 2007).

Other, more mundane, examples of multi-party communications that may use group keys are secure audio and video conferencing, distributed computations, distributed database manipulations and virtually any instance were parties dynamically join and leave the communication stream. These real-world applications have led cryptographic researchers to seek efficient group key protocols for large, dynamic networks comprising several hundreds or thousands of nodes (Burmester and Desmedt, 1995; 1996; Just and Vaudenay, 1996; Steiner *et al*., 1996; Becker and Wille, 1998; Alves-Foss, 2000).

Group key exchange is more properly referred to as group key management, because of the complexities of managing cryptographic keys in a large, dynamic group. Currently, in the literature, there are five main group key management protocols:

- Group Diffie-Hellman (GDH) (Steiner *et al*., 2000)
- Tree-based Group Diffie-Hellman (TGDH) (Kim *et al*., 2000; 2004a)
- Skinny Tree (STR) (Kim *et al*., 2001; 2004b)
- Efficient Group Key (EGK) (Alves-Foss, 2000)
- Communication Computation Efficient Group Key (CCEGK) (Zheng *et al*., 2006)

In general, the goal of these group key management protocols is to optimize either the communication or computation costs associated with dynamic key management, or attempt to balance the communication and computation costs. Communication costs are the size and number of messages (both unicast and broadcast) needed to establish the current group key, while computation costs are the number and extent of calculations (e.g., discrete logarithms or exponentiations) needed to compute or authenticate the group key. Each of the above protocols focuses on improving performance for a specific set of operations or application profile. For example, EGK focuses on key computation costs, while STR focuses on communication costs and CCEGK attempts to balance both communication and computation costs. Communication and computation performance evaluation metrics for group key management operations have been defined in the literature (Zheng *et al*., 2006; 2007).

This study addresses the lack of consistency with respect to dynamic group key management operational definitions and the impact that network View Synchrony (VS) has on those operational definitions and subsequent performance analyses. VS is a specific means of synchronizing every node’s view so that everyone involved eventually has the same view of the network. However, the assumption of wired network VS is improbable for large, dynamic wireless networks and the performance characteristics of group key management protocols change drastically when the assumption is invalidated. Through a defined framework for comparing and contrasting group key management protocols, the importance of VS to group key management protocols, extending into wireless networks, will come into question.

The standard operations necessary for cryptographic key management in dynamic groups need definition, before they can be accurately compared. Once these dynamic group key operations have been explained, the performance metrics used to compare and contrast group key management protocols and show the metrics can be used to gauge the effect of network rebalancing after a massive network change. Then, once VS is described and defined, the assumption of VS will be challenged with regards to use within real-world, wireless ad-hoc networks. This is especially true when framework and metrics defined in this study allow comparison across group key management protocols, whether VS is assumed or not. With the help of definitions and lists of costs, a better working of VS can be understood, especially within the context of the protocols that employ it.

MATERIALS AND METHODS

In order to accurately and fairly compare the four group key management protocols, definitions are needed that explain the operations used by the main group key management protocols. When examining the literature, four full-featured group key management protocols (TGDH, STR, EGK and CCEGK, as GDH is not fully featured), eight major operations were essential for establishing and sharing keys across the dynamic group.

Dynamic group key operations:

- Initialization operation: This is the initial creation
- of the group key and organization of the key management infrastructure

- Join: This operation brings a new member into the existing group
- Mass join (Mass add): This operation allows many new members to be added to an existing group simultaneously when these new members have not already formed a group of their own
- Merge (group fusion): This operation, as opposed to mass join, is used when another group is combined with the existing group to become a new group
- Leave: His operation is used to remove a member from the group
- Mass leave: This operation is used when multiple members are simultaneously removed from the existing group
- Split (partition, or group fission): This operation, different from mass leave, occurs when a single group is divided into two or more component groups
- Key refresh: This operation is to prevent the secret key from being used for a long time. Moreover, to prevent an adversary from breaking in, we should refresh the original key and generate a new secret key periodically

All five existing group key management protocols implement four of the eight basic operations: Join, leave, mass leave and merge. Complications, however, come from two areas. First, some of the protocols do not describe how to implement the rest of the operations: initialization, mass join, split and refresh. As seen in Table 1, GDH and CCEGK are the only protocols that document their implementation of all eight operations. This is important when trying to compare the protocols, to realize that not every operation is implemented in every protocol, so an across-the-board comparison is often not possible. For example, this occurs in TGDH when the authors implement a “split” operation that is called whenever a network partition occurs (Kim *et al*., 2000; 2004a). However, the protocol as described does not consider the cost and behavior of a network that has been partitioned arbitrarily. Rather, their operation simply takes the point-of-view of one group and calculates the cost of removing the partitioned member(s) from it. The rest of the literature considers this to be a classic example of a mass leave (when no consideration is given to the leaving group members). To better clarify, assume there is a group with 1000 members. If a network partition occurred where 900 of those members were isolated, TGDH would calculate the cost of removing those 900 from the initial group and leave it at that. A true split, however, would examine the cost of those 900, either forming a new group of their own, or several smaller groups, which would clearly incur additional unaccounted-for costs. This example clearly demonstrates why operation standardization and agreement is necessary for there to be any meaningful comparison of performance between group key management protocols.

For any considerable evaluation of performance for group key management operations, cost and performance must be quantified. Costs for group key management can be divided into two categories, communication and computation. While historically, one category might be favored over the other, for the limits of this study, neither category is given preference. The following is a list of costs for every operation:

Performance metric for dynamic group key operations:

- Number of rounds: This is a generic time unit used to compare the number of steps taken in different operations. The protocols often require synchronization between rounds, so this number becomes important when taking synchronization time into account
- Number of unicast messages: This is the sum of the number of messages every member sends to other single members in the group per operation. This number is useful for determining total communication and is important if many or all nodes are on the same network collision domain, thus forcing these messages to be sent sequentially rather than simultaneously
- Number of broadcast messages: This is the sum of the number of messages sent by each member to all the other members in the group per operation. Since the messages go to all members of the group, it greatly affects total communication costs depending on the underlying network topology

Table 1: Group key algorithms and their operations

	TGDH	STR	GDH	EGK	CCEGK
Join	X	X	X	X	X
Mass join			X	X	X
Merge	X	X	X	X	X
Split			X	X	X
Initialization			X	X	X
Leave	X	X	X	X	X
Mass leave	X	X	X	X	X
Refresh	X	X	X		X

- Number of messages: This is the sum of the number of unicast messages and broadcast messages. This number is used to determine the total time of communication in an underlying broadcast network
- Number of sequential exponentiations: During an operation there will be a series of computationally expensive cryptographic operations (such as modular exponentiation used in the DH protocol). The protocols in the literature often require the results of one cryptographic operation prior to the execution of another. This metric represented the worst case scenario, the longest sequence of dependencies of these cryptographic calculations in the operation
- Number of signatures: This is the sum of digital signatures used in every round. In every round, the node initiating the operation sends one digital signature
- Number of verifications: Given that each message needs to be verified, the number of verifications is equal to the number of messages; however, several verifications can occur in parallel so care is needed with the number of sequential verifications that must occur during an operation

RESULTS

The only way to reliably compare group key management protocols is to analyze each operation's cost. A tree key structure is often used to internally store keying material for some key management protocols, while others use data structures that behave like linked lists. Each method of data storage has its inherent pros and cons, but the balance of the tree or the size of the linked list is an important factor with respect to efficient operations. For example, after executing a merge, mass add, or mass leave operation, a data structure may be severely unbalanced, which could cause the cost of operations to deteriorate from a logarithmic complexity to linear complexity. Since the number of rounds and sequential exponentiations in nearly all group key management protocols are directly related to the efficiency of the data structure, reducing this cost should be considered essential for operational implementations.

Table 2 excerpted from (Zheng *et al*., 2006) ("Security and performance of group key agreement protocols," a previous paper) illustrates the utility of this study's framework by unifying operational definitions across four group key management protocols and contrasting operational efficiency using the simple metrics defined above.

The definition of View Synchrony (VS) as described by Fekete *et al*. (1997)\ is the basis used with regards to discussing VS for the purposes of this study. In some form or another, every group key management protocol uses VS as a specific means of synchronizing every node's view, ensuring that everyone involved eventually has the same view of the network.

Table 2: Table in comparison of communication

		Communication			
Protocols		Rounds	Messages	Unicast	Broadcast
CCEGK	Initialization	h	2n-2	n	n-2
	Join	1	2	1	1
	Mass join	1	N+1	0	N+1
	Merge	1	N	0	N
	Leave	1	1	0	1
	Mass leave	min (h^i+1, h)	min (2N, n-N)	0	min (2N, n-N)
EGK	Initialization	h	2n-2	0	2n-2
	Join	1	2	0	2
	Mass join	h+1	2N	0	2N
	Merge	N	2N-2	0	2N-2
	Leave	h	(n-1)	0	2(n-1)
	Mass leave	h	2(n-N)	0	2(n-N)
TGDH	Initialization	h	2n-2	0	2n-2
	Join	2	3	0	3
	Mass join	h^i+1	2N	0	2N
	Merge	h^i+1	2N	0	2N
	Leave	1	1	0	1
	Mass leave	min (h^i+1, h)	min (2N, n-N)	0	min (2N, n-N)
STR	Initialization	n-1	2n-2	0	2n-2
	Join	2	3	0	3
	Mass join	2	N+2	0	N+2
	Merge	2	N+1	0	N+1
	Leave	1	1	0	1
	Mass leave	1	1	0	1

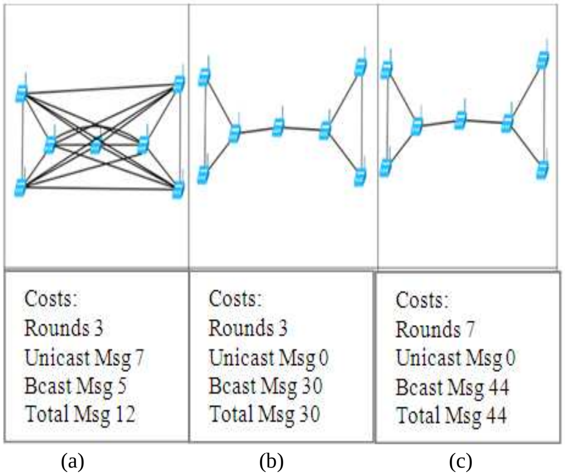

Fig. 1: Side-by-side comparison of protocol initialization, (a) CCEGK with VS and full connectivity (b) CCEGK with VS, without full connectivity (c) CCEK without VS and without full connectivity

Additionally, VS's strength lies in the guarantee that any message sent in a given network view will arrive only at the nodes that are in that network view at the time of sending. This is very useful in measuring the efficiency of group operations, but there are cost overheads for ensuring VS.

The following example shows how the framework and metrics defined in this study allow comparison across group key management protocols, whether VS is assumed or not. Figure 1 shows a side-by-side comparison of the CCEGK initialization operation on a simple 7-node topology with and without full connectivity and with and without VS. Although trivial, the example illustrates how operation performance varies with connectivity and VS Fig. 1a) shows the cost of CCEGK initialization on a fully connected network with VS. Fig. 1b) show CCEGK, as modified to enable initialization on a fully traversable network (but not fully connected) with VS. Fig. 1c) shows the same CCEGK initializing a full traversable network without VS. It is evident that both the number of total rounds transmitted and the time it takes (rounds) is increased when the information is missing that VS makes available (as described previously). As shown and previously described, the loss of VS incurs additional operational costs.

DISCUSSION

As Kim *et al*. (2000; 2001; 2004a; 2004b) describe, VS is used to ensure the fault-tolerance and robustness of the communication and has an expected and accounted for effect on security. However, most authors have considered these costs to be overall negligible compared to the costs of the eight operations. And yet, when the network changes from a traditional wired and fairly static situation to a highly dynamic, potentially mobile ad-hoc network, these costs can no longer be waived aside as negligible. Greater consideration of what VS actually assumes and what it requires must be considered for group key management protocols used in wireless or ad-hoc networks. Still, by and large, it appears that VS can be implemented in wireless networks, but the costs cannot be assumed to be the same as they are in more conventional, wired networks. Many wireless networks are mobile, which requires that all nodes must somehow be informed and updated on their neighbors' locations and status. However, even in non-mobile ad-hoc wireless networks, nodes will often be joining and dropping as network connectivity waxes and wanes. Because this is a far more dynamic and changing environment than traditional wired networks, costs associated with VS are bound to increase dramatically. Furthermore, these costs cannot be ignored and must be addressed in future group key management protocols for wireless ad-hoc networks.

CONCLUSION

Group key management protocol operations are inconsistently portrayed in the literature, making it difficult to compare and contrast the protocols quantitatively. A common framework of operational definitions and performance measures was needed to allow researchers to explore the nuances of protocol differences and better adapt group key management protocols to ad-hoc wireless networks. This study shows how a simple framework of operational definitions and performance measures can be used to quantify differences in protocol operations across network topologies with and without view synchrony.

The framework is neither perfect nor robust, but it does facilitate comparisons between protocols and reduces ambiguity when discussing or describing protocols. Ultimately, better communication in the field of group key management would benefit all parties and ensure that the protocols can be successfully extended to the exciting realm of ad-hoc networks. Researchers interested in exploring group key management in mobile ad-hoc networks need to be cognizant of several practical considerations that are often overlooked by theoretical cryptographers. For example, packet payload size is quite important in low-powered, ad-hoc networks (e.g., mobile sensor networks). Additionally, certain operations (e.g., mass join, split) may result in massively unbalanced data structures that effectively prohibit authentication and key refresh operations in very large networks and yet packet payload and data structure size have generally not been addressed by the existing group key management protocols. These are but two examples of how practical considerations need to be included in future work on group key management protocols.

REFERENCES

Alves-Foss, J., 2000. An efficient secure authenticated group key ex-change algorithm for large and dynamic groups. Proceeding of the 23rd National Information Systems Security Conference, (NISS'00), CiteSeerX, pp: 254-256. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.6947

Becker, K. and U. Wille, 1998. Communication complexity of group key distribution. Proceeding of the 5th Conference on Computers and Communication Security, Nov. 2-5, ACM Press, San Francisco, California, United States, pp: 1-6. http://portal.acm.org/citation.cfm?id=288090.288094

Burmester, M. and Y. Desmedt, 1995. A secure and efficient conference key distribution system. Lecture Notes Comput. Sci., 950: 275-286. DOI: 10.1007/BFb0053443

Burmester, M. and Y. Desmedt, 1996. Efficient and secure conference key distribution. Proceedings of the International Workshop on Security Protocols, Apr. 10-12, Springer-Verlag, London, UK., pp: 119-129. http://portal.acm.org/citation.cfm?id=720375

Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theor., 22: 644-654. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1055638

Fekete, A., N. Lynch and A. Shvartsman, 1997. Specifying and using a partitionable group communication service. Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, Aug. 21-24, ACM Press, Santa Barbara, California, United States, pp: 53-62. http://portal.acm.org/citation.cfm?id=259380.259422

Just, M. and S. Vaudenay, 1996. Authenticated multi-party key agreement. Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security, Nov. 3-7, Springer-Verlag, London, UK., pp: 36-49. http://portal.acm.org/citation.cfm?id=716554

Manz, D., J. Alves-Foss and S. Zheng, 2007. Network simulation of group key management protocols. J. Inform. Assur. Secur., 2: 67-79. http://www.mirlabs.org/jias/manz.pdf

Kim, Y., A. Perrig and G. Tsudik, 2000. Simple and fault-tolerance key agreement for dynamic collaborative groups. Proceeding of the 7th ACM Conference on Computer and Communications Security, Nov. 1-4, ACM Press, Athens, Greece, pp: 235-244. http://portal.acm.org/citation.cfm?id=352638

Kim, Y., A. Perrig and G. Tsudik, 2001. Communication-efficient group key agreement. Proceedings of the 16th International Conference on Information Security: Trusted Information: The New Decade Challenge, June 11-13, ACM Press, Paris, France, pp: 229-224. http://portal.acm.org/citation.cfm?id=510779

Kim, Y., A. Perrig and G. Tsudik, 2004a. Tree-based group key agreement. ACM. Trans. Inform. Syst. Secur., 7: 60-96. http://portal.acm.org/citation.cfm?doid=984334.984337

Kim, Y., A. Perrig and G. Tsudik, 2004b. Group key agreement efficient in communication. IEEE Trans. Comput., 53: 905-921. DOI: 10.1109/TC.2004.31

Steiner, M., G. Tsudik and M. Waidner, 1996. Diffie-Hellman key distribution extended to group communication. Proceeding of the 3rd ACM Conference on Computer and Communications Security, Mar. 14-15, ACM Press, New Delhi, India, pp: 31-37. http://portal.acm.org/citation.cfm?id=238182

Steiner, M., G. Tsudik and M. Waidner, 2000. Key agreement in dynamic peer groups. IEEE Trans. Parall. Distribut. Syst., 11: 769-780. DOI: 10.1109/71.877936

Zheng, S., D. Manz, J. Alves-Foss and Y. Chen, 2006. Security and performance of group key agreement protocols. Proceeding of the IASTED International Conference on Networks and Communication Systems, Mar. 29-31, Chiang Mai, Thailand, pp: 321-327. http://www.csds.uidaho.edu/papers/Zheng06b.pdf

Zheng, S., D. Manz and J. Alves-Foss, 2007. A communication computation efficient group key algorithm for large and dynamic groups. Comput. Networks: Int. J. Comput. Telecommun. Network., 51: 69-93. http://portal.acm.org/citation.cfm?id=1231935