2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force. IEEE. October 2001.
DOI: 10.1109/MILCOM.2001.985773

SECURE GROUP COMMUNICATIONS FOR WIRELESS NETWORKS

B. DeCleene[1], L. Dondeti[2], S. Griffin[1], T. Hardjono[2], D. Kiwior[1], J. Kurose[3], D. Towsley[3], S. Vasudevan[3], C. Zhang[3]

[1]Litton/TASC
Reading, MA

[2]Nortel Networks
Billerica, MA

[3]University of Massachusetts
Amherst, MA

ABSTRACT

In this paper we consider the problem of key management in a highly-mobile wireless networking environment, such as a dynamic, distributed setting in which command and control nodes move along with individual users. In this scenario, data must be securely multicast from one source to many users, requiring that users be properly keyed. Furthermore, because users move in and out of the session (due to mobility, attrition, and reinforcement), in order to preserve confidentiality, it becomes necessary to rekey each time a user enters or leaves. We present a hierarchical framework and key distribution algorithms for such a dynamic environment, with a focus on how keys and trust relationships are transferred when users move between so-called "areas" in the hierarchy. We present several schemes including one that rekeys every time a member moves from area to area and one that delays rekeying so long as security is not compromised. Our preliminary analytical and simulation results indicate that it is possible to trade off communication throughput with computational and security overheads. We also briefly describe a prototype testbed in which we are implementing and experimenting with these algorithms.

INTRODUCTION

Modern military doctrine postulates a dynamic environment where joint US forces are combined with multinational partners to affect rapid and decisive results. As outlined within the Joint Visions 2010 and 2020, achieving this objective requires integration of new technologies that increase the mobility of our armed forces and enable the appropriate real-time information to be shared securely with coalition partners. Under DARPA's Dynamic Coalitions program, researchers are exploring next-generation approaches for managing and sharing sensitive information with partners such that each individual or group is only granted access to information deemed appropriate for the duration that they are participating within the mission.

A critical element in controlling information access is ensuring that only the appropriate individuals have the cryptographic keys that enable them to decode the disseminated information. For example, to maintain forward confidentiality, when a member leaves the session, the remaining members must be rekeyed to ensure that the departing individual cannot listen in on the future communications. Similarly, backward confidentiality requires rekeying when a new member joins an existing session. Otherwise, the new member would be able to decrypt any past archived exchanges for which he/she was not authorized. Since data cannot be exchanged while member's data keys are being updated, the challenge for any key management system is how to generate and distribute new keys such that the data remains secure while the overall impact on system performance is minimized.

Mobility complicates key management by allowing members to not only leave or join a session but also transfer between networks while remaining in the session. Since a mobile user may accumulate information about the local security services for each area he/she visits, the key management system must consider the level of trust to impart to these mobile members and the performance implications should the member leave the session. Furthermore, as a member moves, the network latency between the member and the key management services may change and result in additional performance degradation.

This paper explores the problem of rekeying large numbers of coalition partners where both their access to information and their position vary with time. Given the political necessity that individual coalitions maintain separate networks, we consider both the implication of keying complexity and network topology on the performance of various rekeying approaches.

ALGORITHMS FOR KEY MANAGEMENT

A common approach for designing a scalable network service is to adopt a hierarchical structure, and a number of

recently-proposed key management algorithms have adopted such an approach [4][5][7]. Broadly, these rekeying algorithms operate by hierarchically dividing the key management domain into smaller administratively scoped areas. The details of hierarchical key management differ from one approach to another, and so in our discussion below we adopt a framework based on [5].

Throughout the domain, a *Domain Key Distributor (DKD)* generates the data key used by the session for encrypting the data. The DKD may be collocated with the producer or shared throughout the domain by multiple sessions. As discussed previously, whenever a new member joins a current session or an existing member leaves a session, a new data key must be generated and distributed to ensure both forward and backward confidentiality.

The domain is further divided into disjoint *areas*. An area is unique in that movement within the area does not require any additional signaling with regard to rekeying; and the cost of rekeying members when a join/leave occurs is considered "reasonable." Areas may be small (such as a fine-grained ad-hoc network) or large (such as a satellite broadcast) depending upon the network topology and operational arrangements. Similarly, an area can be either logically or geographically defined.

Within each area, an *Area Key Distributor (AKD)* is responsible for distributing the data key to members within that area. Because the distribution of the data key within an area must itself be secure, area-local keys are used by the AKD to distribute a new data key to members within the area. Approaches for intra-area rekeying include Public Key Infrastructure (PKI), secure multicast, and logical tree-based algorithms such as [1][6][8].

From the definition of area, mobility impacts performance only when members cross between areas. Without AKD reassignment, rekeying messages must cross heterogeneous network boundaries resulting in additional performance degradation. Consequently, member movement between areas requires a coordinated transfer of the security relationships. Inter-area rekeying algorithms address this problem by introducing specific semantics for transferring between areas.

To illustrate, consider two coalition partners providing broadcast services for users in two overlapping geographic areas (see Figure 1). Users moving within each area are managed by their local AKDs and require no coordination between the two coalition broadcasts. On the other hand, when a user crosses from one area into another, then the security relationships must be transferred between the coalition partners through network back channels and then rebroadcast accordingly.

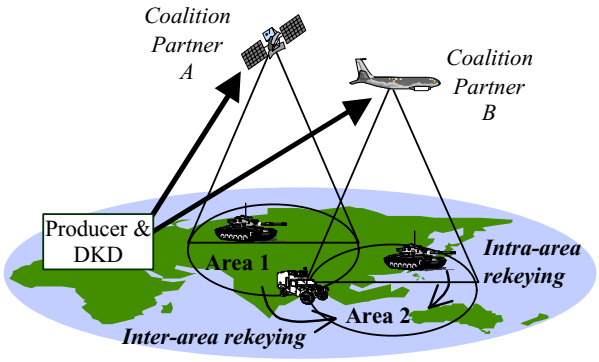

Figure 1. Operational Model

In this paper, we propose multiple inter-area key distribution algorithms, where members may not only enter/leave the session but may also move between areas. These algorithms are defined below.

A. Baseline Rekeying

A direct approach for handling mobility across areas (called the baseline algorithm) is to treat the movement as a leave from the old area followed by a join to the new area. Illustrated in Figure 2, a member leaving the session notifies the local AKD, which halts the current data transmission. Next, the local AKD updates the area key for the remaining members by either securely unicasting to each member using their shared private key, or exploiting a more sophisticated intra-area key protocol such as LKH[6]. Once this is updated securely, a new data key can be distributed to all areas such that the departing member is excluded. At this point, data transmission resumes. This approach ensures forward confidentiality.

During the join, the process is similar. The new member informs the local AKD of its intent to join. Data transmission is halted while a new area key is distributed to the current members (through multicast) and the new member (through unicast). Once complete, the new data key is distributed to all of the members and transmission resumes. This approach ensures backward confidentiality.

The disadvantage of the baseline algorithm is that data transmission is unnecessarily interrupted twice during a transfer between areas because the system cannot distinguish between a departing member and a member that is simply moving. Shown later, the result is degraded throughput and additional computational complexity as extra keys are calculated.

B. Immediate Rekeying

The immediate rekeying algorithm extends the baseline algorithm by adding explicit semantics for a hand-off between areas. Illustrated in Figure 2, the member initiates a transfer by notifying the two affected areas. Each area updates the local area keys per their new membership. However, unlike the baseline algorithm, no new data key is generated and the data transmission continues uninterrupted. Note that when a member actually leaves or joins the session, data transmission is interrupted as new data and area keys are generated per the baseline algorithm described previously.

C. Delayed Rekeying

Both baseline and immediate rekeying algorithms rekey the local areas as soon as a member transfers. As a result, a member that moves rapidly between two areas may cause repeated local rekeying.

Delayed algorithms postpone local rekeying until a particular criterion is satisfied. Members moving between multiple areas may accumulate multiple area keys and reuse these keys when they return to a previously visited area. As always, if a member leaves or joins the session, then the appropriate areas are rekeyed to ensure forward and backward confidentiality.

In pure delayed rekeying, each AKD maintains a list of members that have left the area but still hold valid keys for the area. This is shown in Figure 2. When a member transfers, the area that the member is entering is rekeyed to prevent a member from falsely transferring into an area to get access to the old keys (backward confidentiality). For the departed area, the AKD does not rekey but instead adds the member to the *Extra Key Owner List (EKOL)*. This list is reset whenever a local rekey occurs. When a member returns to an area, it is checked against the EKOL and no new keys are generated if it is on the list.

Extensions to pure delayed rekeying manage the level of information that members outside an area hold. For example, area threshold rekeying causes an area to be rekeyed if its EKOL has more than a particular number of members. Under user threshold rekeying, if a member accumulates more than a fixed number of area keys, then local rekeying occurs. This ensures that no member is able to accumulate all the keys by visiting all of the areas.

D. Periodic Rekeying

Used in conjunction with other inter-area rekeying algorithms, periodic rekeying updates local area keys at particular intervals. This ensures that no local key remains valid for more than a fixed period of time.

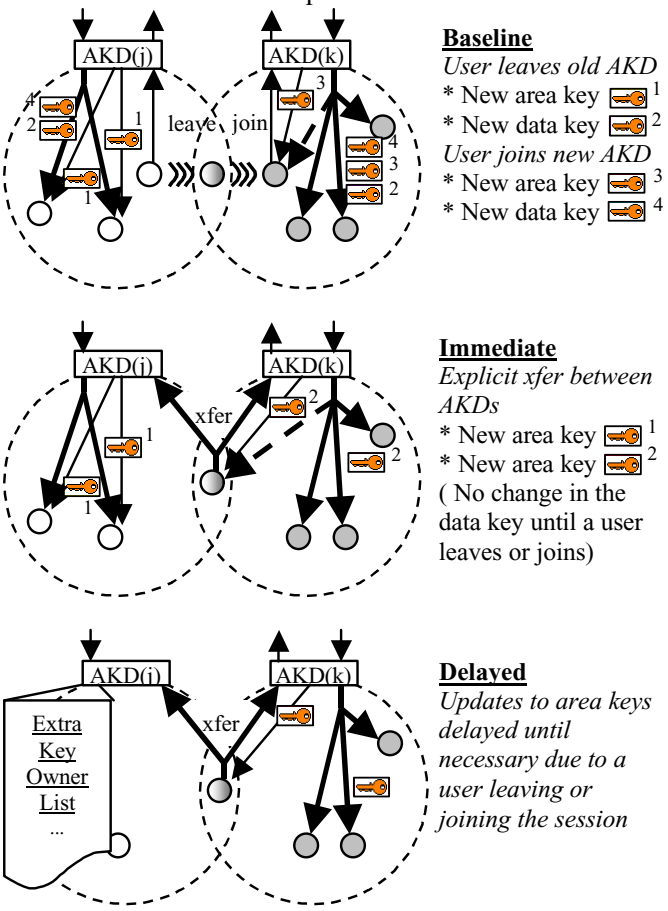

Figure 2. Inter-area rekeying algorithms.

EVALUATION OF REKEYING ALGORITHMS

We describe a preliminary evaluation of the four inter-area rekeying algorithms described earlier. Specifically, we compare their performance and the level of trust imparted to the users in the system (i.e. security). We consider the following metrics

- Rekeying rates (R_d, R_a) measures the rates at which data and area keys are generated respectively.
- Mean number of extra-keys (K_m, K_a) measures the average number of valid extra keys held by a member outside the area; and the average total number of valid keys held by all members outside the area.
- Percentage of time off-line (P_o) due to rekeying measures the end-to-end throughput degradation.

Our evaluation assumes that the network behavior does not change over time. We assume that there are M areas in the system, that members arrive to the session with rate λ, and that they are equally likely to select an area. Once inside

the system, a member spends an average amount of time in an area equal to $1/\mu_a$. Mobility in our model is captured by the average number of areas visited by a member before leaving the system, X. Last, let T denote the period between rekeys in the periodic rekeying algorithm.

The behavior of all of the algorithms, with the exception of the pure delayed rekey algorithm, is depicted in Figure 3. Clearly, the baseline performs poorly whenever there is any amount of mobility, i.e., $X > 1$. For example, the immediate rekey algorithm reduces the data rekey rate without either adding overhead or compromising security.

	R_d	R_a	K_m	K_a
baseline	$2X\lambda$	$2X\lambda/M$	0	0
immediate	2λ	$2X\lambda/M$	0	0
delayed	2λ	$< 2X\lambda/M$	> 0	> 0
periodic	2λ	2λ/M+1/T	0	0

Figure 3. **Performance Comparison of Rekeying Algorithms**

It is difficult to evaluate either the delayed rekey algorithm or to determine the percentage of time off line, P_o, without making additional assumptions. In [9] we developed an analytical model of the delayed rekey algorithm under the assumption that arrivals are described by a Poisson process, that time spent in an area is exponentially distributed, and that members traverse areas in a probabilistic manner. Based on the analysis reported in [9], we compare the immediate rekey algorithm with the delayed rekey algorithm.

In Figure 4, we plot the ratio of the delayed area rekey rate to the immediate rekey rate as a function of the average number of areas visited by a member. When there is little mobility, there is little difference between an immediate rekey algorithm and a delayed rekey algorithm. As mobility increases, the differences between the two approaches increase, especially for the case that λ is small.

In Figure 5, we plot the average number of area keys that are held by members outside the area under the delayed rekey algorithm, K_m. As one might expect, we observe that the measure of insecurity increases as a function of the arrival rate, λ, and the mobility, X. However, due to frequent rekeying due to member joins and leaves, the average number of keys outside the area is small.

Last, we focus on P_o, the percentage of time off-line. We report results obtained from simulation for this measure. Figure 6 illustrates the behavior of P_o, as a function of mobility, specifically X, the average number of areas visited per member, for the three rekeying policies as well as no rekeying. As expected, both no rekeying and baseline do very poorly compared to the immediate and

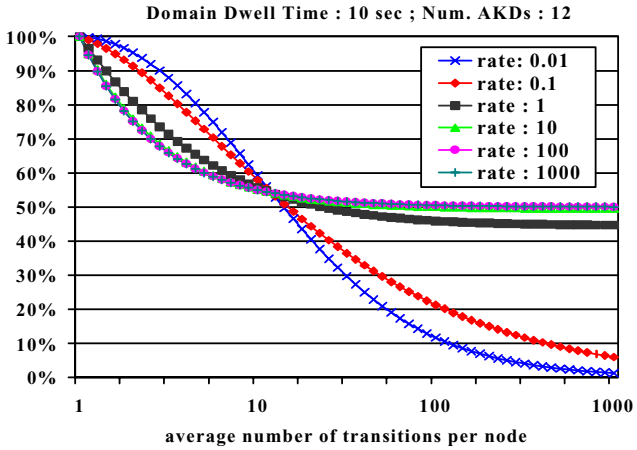

Figure 4. Ratio of Delayed and Immediate Rekey Rates

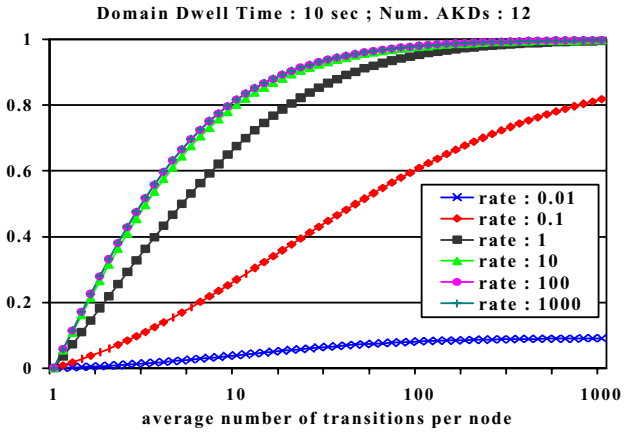

Figure 5. Average Number of Members Outside Area w/ Key

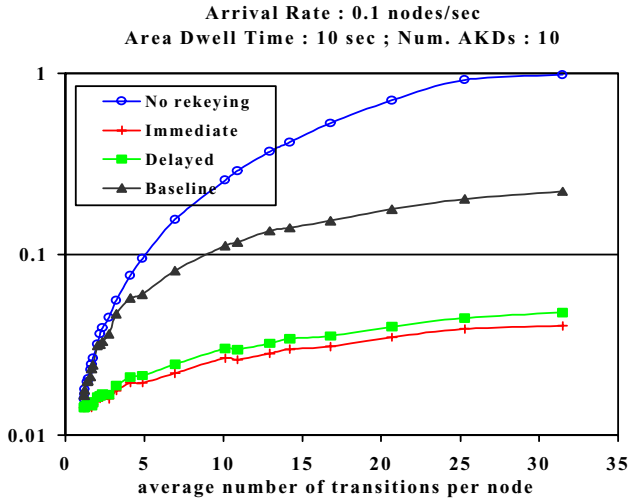

Figure 6. Percentage time off-line due to rekeying

delayed rekey policies. Surprisingly, immediate rekeying outperforms delayed rekeying. This is because delayed rekeying must rekey multiple areas when the member departs rather than a single area. Since the data transmission cannot resume until all the visited areas have been rekeyed and a new data key distributed, the delayed algorithm remains off-line longer than the immediate algorithm. Logical tree-based intra-area algorithms help reduce the time required to each area and thereby reduce the difference between the two algorithms.

KEY MANAGEMENT TESTBED

TASC, together with its subcontractors, has established a key management testbed for conducting experiments on different rekeying algorithms. Similar to Figure 1, the testbed consists of two separate 802.11 wireless networks connected through a WAN emulator. An AKD is located within each area and supports an arbitrary number of members. By configuring the WAN emulator to represent different delay and loss characteristics, a wide-variety of dynamic coalition scenarios can be explored such as an in-theater satellite broadcast.

Coalition members are modeled using multiple wireless laptops running Linux v6. These laptop computers migrate between the two areas according to mobility models that are defined a-priori. Depending upon the experiment being performed, these mobility models may be based on stochastic properties of the members or actual recorded position information for in-theater users.

Exploiting a common overall structure, support for various rekeying algorithms is being developed as part of a Key Management Framework (KMF). Based loosely on the GSAKMP daemon [3], KMF provides a flexible framework and supporting toolkit for developing intra-area and inter-area rekeying algorithms. Elements of the toolkit include encryption and authentication services as well as support for integrating with extant military applications. An underlying communications package provides both reliable unicast and multicast for disseminating keys to members and other key management distributors. This is used to extend the key management services to support both in-band and out-of-band rekeying.

CONCLUSIONS

Under the DARPA/ATO Dynamic Coalitions program, rekeying strategies are being developed to support next-generation multi-national mobile troops. As the number of members and their mobility increases, the performance of these algorithms can have significant impact on the end-to-end performance of the system. By introducing explicit semantics for managing user mobility between disparate areas, substantial improvements in performance have been demonstrated. Additional improvements can be obtained by deferring local rekeying until a member departs or joins the session. For highly mobile users who remain in the session for a reasonable duration of time, delayed rekeying can provide significant improvements by reducing the number of keys generated and distributed. Through a combination of analysis, simulation, and implementation, this research extends our information superiority beyond today's capabilities.

ACKNOWLEDGMENT

This research was funded, in part, by DARPA/ATO's Dynamic Coalition program under contract N66001-00-C-8011.

REFERENCES

[1] D. Balenson, D. McGrew, A. Sherman, "Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization," draft-irtf-smug-groupkeymgmt-oft-00.txt, September 2000, Work in Progress.

[2] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, D. Saha. "Key management for secure Internet multicast using boolean function minimization techniques", In Proceedings of IEEE Infocom, New York, USA, March 1999.

[3] GSAKMP Daemon software, Sparta, Inc. ftp.sparta.com/pub/columbia/gsakmp, July 2000.

[4] T. Hardjono, B. Cain and N. Doraswamy, "A Framework for Group Key Management for Multicast Security," Internet Draft, draft-ietf-ipsec-gkmframework-03.txt, August 2000, Work in Progress.

[5] T. Hardjono, B. Cain, I. Monga, "Intra-Domain Group Key Management Protocol," Internet draft, draft-ietf-ipsec-intragkm-01.txt, July 1999, Work in Progress.

[6] H. Harney and E. Harder, "Logical Key Hierarchy Protocol," Internet draft, draft-harney-sparta-lkhp-sec-00.txt, March 1999.

[7] S. Mittra, "The Iolus Framework for Scalable Secure Multicasting," In Proceedings of ACM SIGCOMM, Cannes, France, September 1997.

[8] C. K. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," In Proceedings of ACM SIGCOMM, Vancouver, Canada, August-September 1998.

[9] C. Zhang, B. DeCleene, J. Kurose and D. Towsley, "Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications" Submitted to NGC2001.