A TIME DRIVEN METHODOLOGY FOR KEY DIMENSIONING IN MULTICAST COMMUNICATIONS*

Roberto Di Pietro, Luigi V. Mancini, and Alessandro Mei
Dipartimento di Informatica, Università di Roma "La Sapienza",
Via Salaria 113, 00198-Roma, Italy.
{dipietro, mancini, mei}@dsi.uniroma1.it

Abstract This work considers the key management for secure multicast in the Logical Key Hierarchy (LKH) model, and proposes a methodology to establish the minimal key length that guarantees a specified level of confidentiality. We reach such a result by analyzing and extending the threat model to the confidentiality of the multicast information. For this extended threat model, we present a methodology that takes into account the following parameters: (1) the required lifetime of the information confidentiality; (2) the level of the key in the LKH model; (3) the dynamics of the multicast group, that is the eviction rate of the users. From these rationales we develop an analytical model that, for each level, derives the appropriate key length, that is the minimal length that assures the desired degree of confidentiality under the hypotheses in the threat model. Finally, for a specific instance of the LKH model, we describe a numerical example that shows the saving that can be achieved in terms of the key lengths.

Keywords: key management protocol, time driven methodology, threat model, secure multicast, reliable multicast, efficient multicast.

1. Introduction

Many emerging applications (for instance pay per view TV, stock option bid), are based upon a group communication model. In particular, they require message delivery from one or more authorized senders to a large number of authorized receivers. This model is available on the Internet, where multicast communication has been successfully imple-

*This work was partially supported by the EU IST-2001-34734 EYES project and by the Italian MIUR under the WEB-MINDS project.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35691-4_52

D. Gritzalis et al. (eds.), *Security and Privacy in the Age of Uncertainty*
© IFIP International Federation for Information Processing 2003

mented to provide an efficient, best effort delivery service to large groups of users Dee88. The deployment of network applications requiring group communication will accelerate in coming years. Thus, security is an important concern to enable the adoption and diffusion of the multicast paradigm.

Cryptographic techniques should prevent unauthorized users from accessing the content of delivered messages. As a result, securing group communications, that is, providing confidentiality, authenticity, and integrity of messages delivered between group members, is a critical issue. In this paper, message confidentiality is assured by using simple and efficient private key techniques for group data encryption. Note that, although group communications using sophisticated cryptographic techniques are proposed in the literature STW98, our choice is motivated by the fact that asymmetric cryptography requires much more resources to decrypt messages Sch96, thus depleting the level of service perceived by the users.

Confidentiality is among the main requirements we consider: only authorized users should decrypt a multicast message, even though this message is broadcast over a geographical region. The confidentiality requirement can be translated in the context of secure multicast into the following four requirements on key distribution: (1) Non-group Confidentiality: users that were never part of the group should not have access to any key that can decrypt any multicast data sent to the group; (2) Forward Confidentiality: users deleted from the group at some time t do not have access to any key used to encrypt data after t, unless they are authorized to join again the group; (3) Collusion Freedom: no subset of deleted users should be able to decrypt future group communications, even by sharing the keys they had before deletion; (4) Backward Confidentiality: a user added at time t should not have access to any key used to encrypt data before t while the user was not part of the group.

This paper focuses on key distribution schemes for secure group communications. In particular, we focus our efforts on an efficient use of the bandwidth available to the center. The available bandwidth is one of the most valuable resources affecting the management of group dynamics ASW00; CMN99. To reduce the bandwidth used by management messages, we derive a model in which the length of the encryption keys is reduced with respect to the current solution in literature, while not affecting the overall security of the model.

We provide a methodology to select the bit length of the encryption keys in the LKH model. In particular, we provide: (1) a threat model for information confidentiality in multicast communications; (2) an analytical model which supports the following results: (a) the length of the encryption keys can be set accordingly to the parameters indicated in the threat model and the key level in the LKH hierarchy; (b) such a key length is shorter than the key length from current literature, without weakening the security level enforced by the standard model; (c) the key length decreases as the number of users in the system increases; (d) our methodology achieves savings in the number of bits to be broadcast of the order of $\Theta(\log^2 n)$ with respect to current re-keying schemes (see Section 1.5), and these savings are remarkable in real-world numerical examples as well. Reducing the key length in the LKH model, reduces the re-keying completion time, enhance the reliability of the key transmission and reduces the storage requirement for the users.

In Section 1.2, we describe the multicast communication model assumed in this paper, while in Section 1.3 we define the threat model. In Section 1.4 we first derive an analytical methodology, and then we show a numerical example. In section 1.5 we describe related work in the field. Finally, Section 1.6 contains some concluding remarks.

2. Multicast Model

2.1 The LKH Model

We assume the LKH model described in WGL00 (*key graph model*). That is, there is a multicast group $\mathcal{M} = \{u_1, .., u_n\}$, which is a dynamically changing subset of all possible users. The set can change according to the eviction of a user from the group, or according to a new user joining the group. We assume there is a super user, called *the center*, which can send a message to the multicast group that can be received by all members of $\mathcal{M}$. The message is sent over an insecure channel, therefore the same message can be received by other entities not belonging to $\mathcal{M}$. To enforce the privacy of data, we can assume available to the users in $\mathcal{M}$ and to the center a cryptographic module, based on symmetric key cryptography. In the following, when the center needs to send a message m to $\mathcal{M}$, the center will compute $m' = E_k(m)$ and then will broadcast to the group $\mathcal{M}$ the message. The key k, shared by all users and the center is the session key. A join occurs when a new user u_j is added to $\mathcal{M}$, while a deletion occurs when a user u_e is evicted from $\mathcal{M}$.

The *key graph* requires to build up a tree of auxiliary keys (in the following *key graph tree*), whose leaves are the private key of the users in $\mathcal{M}$. Each user has to store the keys that are on the leaf-root path. Within this framework, an excellent trade-off between the *per user* required storage, and efficiency in the number of message required is achieved.

Referring to this model, we will adopt the following terminology: (1) the siblings of a user u_i in the key graph is the set of users that share

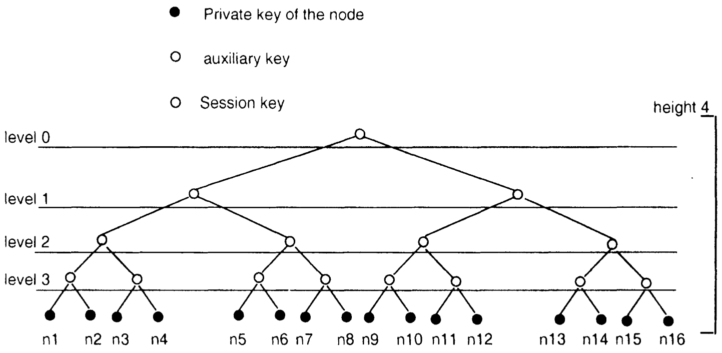
● Private key of the node
○ auxiliary key
○ Session key
height 4
level 0
level 1
level 2
level 3
n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13 n14 n15 n16

Figure 1. Notation and terminology employed in the following.

with the user u_i the same parent; (2) we will assume, unless otherwise specified, that the number of the users in the key graph is exactly $n = |\mathcal{M}| = b^h$, where b is the ariety of the key graph. This assumption (that is, the key graph is perfectly balanced) does not affect the generality of our findings, but simplifies the presentation; (3) the root of the key graph (that is the session key) is located at height 0, while its leaves are at height h; (4) the user to be excluded from the multicast group will be denoted by u_e. In Figure 1 an example of such a key tree. Note that the relevant literature in the field assumes that all keys in the key graph tree have the same bit length.

In the following, we assume, without loss of generality, that $b = 2$. That is, the key graph we consider is a binary tree. However, the same kind of results hold for an arbitrary tree ariety.

2.2 Security in the LKH Model

When the session key has to be changed due to the occurrence of an eviction, all users need to change the auxiliary keys shared with the evicted user u_e. Indeed, employing one of these keys to encrypt any message would result in a violation of the privacy, since the encryption key is hold by u_e, which could correctly decrypt the message. When a join occurs, the newly joined user u_j receives the appropriate sequence of auxiliary keys from the center. The center encrypts such messages *via* the private key of u_j, which is known only by the center and u_j himself.

It has been shown in CGI$^+$99; YLZL01 that the most complex re-keying operation is an eviction, and in the following we will deal with this kind of re-keying operation. Moreover, a re-keying operation due to a join can be performed according to an optimal algorithm described in WCS$^+$99.

3. The Threat Model and its Implication

Information confidentiality is a time related property. Indeed, information usually requires to be protected from unauthorized access for a certain period of time only, after which it is not worth securing that information any longer. For instance, stock quotes may need to be protected for, say, twenty minutes. After such a period, this information can be assumed to be publicly known, and there is no point in assuring its confidentiality. We refer to the period during which it is worth securing a given information as its *lifetime* (LL). A first ingredient to the definition of the lifetime of a piece of information is the period t_d for which the manager of the multicast group desires this information to be confidential. In general, $LL \leq t_d$, as it is worthless to enforce confidentiality beyond t_d. However, LL and t_d are usually different. For instance, even though a broker in the stock market could desire to keep stock quotes confidential for one hour, after twenty minutes this information is uncovered by many different sources. Hence, while t_d is equal to 60 minutes, the *lifetime* cannot be more than 20 minutes.

Given the goal of enforcing confidentiality of a piece of information throughout its lifetime, it is important to define the capabilities of the attackers. In the next section, we will describe the threat model for multicast communications in a dynamic group of users.

3.1 The Threat Model

Our threat model assumes that an enemy has the capability of intercepting all the communications between the center and the multicast group. That is, the enemy is able to eavesdrop all the encrypted payload traffic, as well as the management traffic. The management traffic is the set of communications which do not carry payload information, like rekeying messages. However, we limit the adversary to be passive, that is the adversary cannot inject forged messages in the communication channel. This assumption is coherent with our goal to preserve information confidentiality.

We extend such a threat model by taking into account the behavior of the currently legitimate users, as well as the behavior of the evicted users. As for the legitimate users, we assume that a legitimate user does not leak any information he holds. This assumption is congruent with the relevant literature of the field (see Section 1.5). Indeed, to discourage any legitimate user from disclosing its keys, a tracing traitor scheme

NNL01; BPS00 can be applied. Note that such a solution does not prevent a legitimate user to leak its keys, but can deter users from such a malicious behavior since the leaking user can be traced and prosecuted.

As for evicted users, we assume that each evicted user can have a malicious behavior. In particular, we adopt the conservative assumption that a coalition of evicted users can cooperate to recover the new keys employed by the center to securely communicate to the group. Moreover, we assume that as soon as a user u_e is evicted from the multicast group, u_e can disclose all the previous received information.

4. Choosing the Key Length

Even without knowing the key used to encrypt a message, the attacker can recover its plain text. The time required by this operation is related to the encryption mechanism and the length of the encryption key. A number of techniques aimed at recovering the plain text from the cypher text exist Sch96, but their analysis is out of the scope of this paper.

These techniques are available to the attacker (an evicted user as well as an external entity). Given his computational capabilities, it is possible to find out the time required by the attacker to recover from the cypher text the corresponding plain text. We assume that this time depends exponentially with the key length, that is, adding one bit to the encryption key yields a message requiring twice as much time to be broken.

4.1 The Session Key

The session key protects the payload. Therefore, its bit length should be related to the confidentiality requirements of the payload, that is, its lifetime. A first upper bound on the lifetime of the payload can be given by the manager of the multicast service. Parameter t_d measures how long the information sent is desired to be protected. However, this is not the only factor. Note that, once a legitimate user is evicted from the multicast group, we are not guaranteed that the same user does not leak either the past session key or the payload itself to the public. Consequently, there is no point in protecting past payload held by an evicted user, since we can assume this payload to be compromised. The above remarks can be summarized in the following statement, where t_e refers to the time interval between one eviction and the next one.

Statement 1. *The lifetime LL of the payload depends on the desired level of confidentiality t_d of the information as well as on the frequency of the evictions t_e of the users in the multicast group, according to the formula $LL = min\{t_d, t_e\}$.*

The session key has to resist from malicious attacks for time LL. Assuming that the kind of an attacker we are willing to stop is capable of breaking an s bit key in time t_q, the length of the session key k_0 should be set to:

$$|k_0| = s + \lceil \log(LL/t_q) \rceil. \qquad (1)$$

The above discussion does not hold any more when considering an auxiliary key in the key graph tree. Take, as an example, one of the two keys at level 1. Such a key is not changed at every user eviction, and, when it is not changed, is used to multicast the new session key. It is thus intuitive that this key should be stronger than the session key, since it has to resist for a longer time interval.

Statement 2. *The length of an auxiliary key in the key graph tree depends on the desired level of confidentiality t_d of the payload, as well as on the frequency of the evictions t_e of the users in the multicast group, and on the position of the key in the key graph tree.*

¿From the above discussion it follows that the length of the keys in the LKH model does not need to be the same for all the keys in the tree. In the next section, we will show an analytical model to formalize the above discussion.

4.2 The Auxiliary Keys

In the following, we assume that the eviction probability of every single user u_i is exponentially and independently distributed with parameter λ. That is, the probability that user u_i is evicted within time t is given by: $\mathcal{P}(u_i \leq t) = 1 - e^{-\lambda t}$. Note that, due to the exponential law properties, the probability that at least one eviction occurs from the multicast group $\mathcal{M}$ within a period t is given by: $\mathcal{P}(\mathcal{M} \leq t) = 1 - e^{-n\lambda t}$. On the average, one user is evicted every $t_e = 1/(n\lambda)$ time, which is the expectation of an exponentially distributed random variable with parameter $n\lambda$.

Consider an intermediate key $k_{i,j}$, where i is the level and j is the position within the level from left to right in the key graph tree. Let $S_{k_{i,j}}$ be the set of users sharing key $k_{i,j}$. Key $k_{i,j}$ can be considered as the root of a subtree having $|S_{k_{i,j}}| = 2^{h-i}$ leaves. When an eviction occurs, the probability that the evicted user u_e belongs to the set $S_{k_{i,j}}$, for all $i \in [0 \ldots h-1]$ and for all $j \in [0 \ldots 2^{h-i} - 1]$, is given by:

$$\mathcal{P}(u_e \in S_{k_{i,j}}) = \frac{|S_{k_{i,j}}|}{n} = \frac{2^{h-i}}{n}; \qquad (2)$$

that is the position of u_e is uniformly distributed over the leaves of the key graph. ¿From Equation (2), we can derive the following straightforward fact.

Fact 1. *For any two keys k_{i,j_1} and k_{i+1,j_2} the probability that the evicted user u_e holds key k_{i,j_1} is twice the probability that u_e holds key k_{i+1,j_2}.*

The above framework allows to set a probabilistic methodology for choosing the length of each key in the key graph according to our threat model. Note that, in our model, assuming that all the keys have the same length $L_{\max}$ is counterintuitive. Indeed, the lower the level of a key, the higher the number of users sharing that key and therefore the higher is the probability of that key to be changed due to an eviction. For instance, consider the root key: at each eviction such a key is renewed. Consequently its lifetime is shorter than all other keys, and its length can be reduced as follows without compromising in any way the confidentiality of the protocol.

¿From Fact 1, keys at level i, with $i \in [0 \ldots h-1]$, should be asked to be $L_{\min} + i$ bits long, where $L_{\min} = |k_0|$. Hence, the total number of bits on a leaf-root path to broadcast when an eviction occurs is:

$$\sum_{i=0}^{h-1} |k_i| = \sum_{i=0}^{h-1} (L_{\min} + i) = hL_{\min} + \sum_{i=0}^{h-1} i = hL_{\min} + \frac{(h-1)h}{2} \tag{3}$$

Therefore, the keys of maximal length are those at level $h-1$, whose length is $L_{\max} = L_{\min} + (h-1)$ bits.

Now focus on the standard key graph, in which all the keys are of equal length. We want to investigate what key length assures the same level of confidentiality provided by our methodology. It is easy to realize that the length of all the keys must be at least $L_{\max}$. Indeed, choosing a shorter length weakens the keys at highest levels, and thus compromises the strength of the whole scheme. For a standard re-keying operation, when all keys have the same length $L_{\max}$, the total number of bits of the keys to be changed is:

$$\sum_{i=0}^{h-1} |k_i| = \sum_{i=0}^{h-1} L_{\max} = hL_{\max}. \tag{4}$$

Thus the fraction of bits saved employing our methodology is given by:

$$\left(hL_{\max} - \left(hL_{\min} + \frac{h(h-1)}{2} \right) \right) / hL_{\max} = \frac{h-1}{2L_{\max}}. \tag{5}$$

Note that the saving in the number of bits increases as the number of users in the system increases.

4.3 A Numerical Example

In this subsection we show with a numerical example the possible savings that can be achieved in the key length by adopting our methodology. In this example, the system is characterized as follows:

- the multicast group consists of $n = 2^{16} = 65,536$ users;
- each user leaves $\mathcal{M}$ according to the exponential law of parameter $\lambda = 1/180$, where time is measured in days. This essentially means that each user belongs to the multicast group for 180 days on the average;
- the desired confidentiality lifetime of the information t_d is equal to 15 days;
- to recover 20 bits of plain text from 20 bits of cypher text, without any knowledge of the encryption key, the attacker takes 10 seconds.

¿From these hypotheses, a user is evicted from $\mathcal{M}$ approximately every 238 seconds on the average. According to the threat model, the lifetime can be computed as the minimum between the desired period of confidentiality and the eviction rate, that is LL is equal to 238 seconds. Consequently, key $k_{0,0}$ should be $20 + \lceil \log(238/10) \rceil = 25$ bits long. According to Fact 1, all keys $k_{i,j}$, for all $i \in [1 \ldots h-1]$ and $j \in [0 \ldots 2^i - 1]$, should be $25 + i$ bits long. Therefore the total length of all keys to be changed when an eviction occurs is, by Equation 3, equal to $hL_{\max} - (h-1)h/2 = 520$ bits. Note that a re-keying operation within the classical LKH model would have required, by Equation 4, $hL_{\max} = 16 * (25 + 15) = 640$ bits. The saving achieved is equal to the 18.75%. Indeed, by Equation 5, $(h-1)/2L_{\max} = 15/(2 * 40) = 18.75$.

4.4 Comparison

In this subsection, we show the gain that can be achieved by employing our methodology with respect to the other protocols in the literature. We refer to the protocol proposed in this paper as Time Driven Key Dimensioning (TDKD). For a detailed comparison of current protocols, not including TDKD, see PMJ02. Due to room limitation, we show the total number of bits sent by the center in case of a single user eviction and a single user join only (see Table 1). With K, we refer to the size of the key of maximal length L_{max} in bits, and with I to the user ID ($\log n$ bits long).

Model	Join: message size			Leave:Multicast Message size
	Join unic.	Sibling unic.	Multicast	
LKH++	$(h+1)K$	0	$K+I$	$I+(h-1)K$
EHBT	$(h+1)K$	I	hI	$I+hK$
ELK	$(h+1)K$	I	0	$I+h(n_1+n_2)$
OFT	$(h+1)K$	$I+2K$	$(h+1)K$	$I+(h+1)K$
LKH+	$(h+1)K$	$I+K$	hI	$I+2hK$
TDKD	$h(K-\frac{h-1}{2})$	0	$K+I$	$I+h(K-\frac{h-1}{2})$

Table 1. Comparison of messages size for a single user joining or leaving $\mathcal{M}$.

5. Related work

The Logical Key Hierarchy (LKH) model was introduced in HH99; WHA99; WGL00 to manage multicast group within a framework of security and efficiency. In WCS$^+$99 a protocol is proposed called LKH+. The performances achieved by LKH+, as for the join of new users, improve on WGL00. Indeed, the keys to be refreshed, instead of being encrypted with their respective children's keys, are refreshed by hashing.

The OFT protocol MS98; BMS02 requires each re-keying message to carry just $\log n$ keys instead of $2\log n$ as in the basic LKH model. The ELK protocol PST01 focuses on reliable and secure multicast for large groups, providing a periodic re-keying scheme. The ELK protocol is similar to OFT, but ELK employs pseudo random functions to build and manipulate the keys in the tree. In RMH01, the use of the old keys to create or update the new keys saves information to be transmitted to the users when a membership change occurs.

In PMJ02, protocol LHK++ is introduced. This protocol takes advantage of: (1) the set of information the users share that can be used to locally generate the new keys; (2) the set of keys that the users logically share from a certain point onward in the LKH, and that allows the users to compute locally their path to the root employing a one-way hash function. In this way, it has been shown how to reduce of 50% the bandwidth required by the center for unicast communications to perform group set-up, and to deal with mass join. Moreover, this approach allows to save more than 50% of the computations required by the center in mass evictions and requires less computations on the user device than the other solutions.

About reliability in multicast communication, the protocol proposed by Yang et al. YLZL01 uses proactive FEC in which parity packets are transmitted along with payload packets in each FEC block. An interesting approach proposed by some works KCWP00; TSPL01 is to send the key updates in the same stream as data packets. The main advantage is that key updates are synchronized with the encrypted data, and a separate protocol is not needed for reliable key delivery.

6. Concluding Remarks

In this paper, we have described a methodology to establish the key length of the encryption keys in the LKH model. In particular, we have developed an analytical model, based on a threat model for information confidentiality in multicast communications, that proves the length of the encryption keys can vary according to: (1) the eviction rate of the user in the multicast group; (2) the level in the key graph tree at which the logical key is placed; (3) the desired level of confidentiality. Note that our model provides two properties: first, reducing the length of the encryption keys does not weaken the security of the LKH model; second, the length of the key decreases as the number of users in the system increases. Reducing the keys length in LKH reduces the re-keying completion time, enhances the reliability of the key transmissions and reduces the storage requirement for the users.

References

[ASW00] M. Abdalla, Y. Shavitt, and A. Wool. Key management for restricted multicast using broadcast encryption. *IEEE/ACM Transaction on Networking*, 8(4), 2000.

[BMS02] D. Balenson, D. McGrew, and A. Sherman. Key management for large dynamic groups: One-way function trees and amortized initialization. Internet draft, IETF, June 2002.

[BPS00] Omer Berkman, Michal Parnas, and Jiri Sgall. Efficient dynamic traitor tracing. In *Symposium on Discrete Algorithms*, pages 586–595, 2000.

[CGI$^+$99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: a taxonomy and some efficient constructions. In *Proceedings of IEEE INFOCOM'99: Conference on Computer Communications*, 1999.

[CMN99] R. Canetti, T. Malkin, and K. Nissim. Efficient communication-storage tradeoffs for multicast encryption. In Springer-Verlag, editor, *Advances in Cryptology, EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, 1999.

[Dee88] S. E. Deering. Multicast routing in internetworks and extended LANs. *Computer Communication Review*, 18(4), 1988. ACM SIGCOMM '88 Symposium: Communications Architectures and Protocols.

[HH99] H.Harney and E. Harder. Logical key hierarchy protocol. Internet draft, IETF, April 1999.

[KCWP00] M. Kandansky, D. Chiu, J. Wesley, and J. Provino. Tree-based reliable multicast (tram). IETF Internet Draft, 2000.

[MS98] D. A. McGrew and A. T. Sherman. Key establishment in large dynamic groups using one-way function trees. Technical Report 0755, TIS Labs at Network Associates, Inc., Glenwood, MD, May 1998.

[NNL01] Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. *Lecture Notes in Computer Science*, 2139:41–62, 2001.

[PMJ02] R. Di Pietro, L. V. Mancini, and S. Jajodia. Efficient and secure keys management for wireless mobile communications. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 66–73. ACM Press, 2002.

[PST01] A. Perrig, D. Song, and D. Tygar. ELK, a new protocol for efficient large-group key distribution. In *Proc. of 2001 IEEE Symposium on Security and Privacy*, pages 247–262, 2001.

[RMH01] S. Rafaeli, L. Mathy, and D. Hutchison. EHBT: an efficient protocol for group key management. *Lecture Notes in Computer Science*, 2233:159–171, 2001.

[Sch96] B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C.* John Wiley & Sons, 1996.

[STW98] M. Steiner, G. Tsudik, and M. Waidner. CLIQUES: A protocol suite for key agreement in dynamic groups. In *Proceedings. 18th IEEE International Conference on Distributed Computing Systems*, 1998.

[TSPL01] W. Trappe, Jie Song, R. Poovendran, and K.J.R. Liu. Key distribution for secure multimedia multicasts via data embedding. In *Proc. of IEEE ICASSP 2001*, pages 1449–1452, 2001.

[WCS$^+$99] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner. The versakey framework: Versatile group key management. *IEEE Journal on Selected Areas in Communications*, 17(9):1614–1631, September 1999.

[WGL00] C. K. Wong, M. Gouda, and S. S. Lam. Secure group communications using key graphs. *IEEE/ACM Transaction on Networking*, 8(1), 2000.

[WHA99] D. Wallner, E. Harder, and R. Agee. Key management for multicast: Issues and architectures. RFC 2627, June 1999.

[YLZL01] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam. Reliable group rekeying: a performance analysis. In *Proc. of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 27–38. ACM Press, 2001.