Energy-Efficient Group Key Agreement for Wireless Networks

Thomas R. Halford, *Member, IEEE*, Thomas A. Courtade, *Member, IEEE*, Keith M. Chugg, *Fellow, IEEE*, Xiaochen Li, and Gautam Thatte

***Abstract*—Advances in lattice-based cryptography are enabling the use of public key algorithms (PKAs) in power-constrained ad hoc and sensor network devices. Unfortunately, while many wireless networks are dominated by group communications, PKAs are inherently unicast—i.e., public/private key pairs are generated by data destinations. To fully realize public key cryptography in these networks, lightweight PKAs should be augmented with energy-efficient mechanisms for group key agreement. We consider a setting where master keys are loaded on clients according to an arbitrary distribution. We present a protocol that uses session keys derived from those master keys to establish a group key that is information-theoretically secure. When master keys are distributed randomly, our protocol requires $O(\log_b t)$ multicasts, where $1 - 1/b$ is the probability that a given client possesses a given master key. The minimum number of public multicast transmissions required for a set of clients to agree on a secret key in our setting was recently characterized. The proposed protocol achieves the best possible approximation to that optimum that is computable in polynomial time. Moreover, the computational requirements of our protocol compare favorably to multi-party extensions of Diffie-Hellman key exchange.**

***Index Terms*—Ad hoc wireless networks, public key cryptography, wireless sensor networks.**

I. Introduction

Security against malicious eavesdroppers is a paramount concern in wireless networks. While symmetric key algorithms provide a lightweight means of ensuring data confidentiality in power-constrained devices, they may be ill-suited to applications where the devices can be compromised. For example, if a single client in a sensor network employing AES-256 with a common key is compromised, then all of the other clients must be rekeyed. The transmissions required by over-the-air rekeying of an entire wireless network can consume significant energy. The use of public key algorithms (PKAs) can in principle address this issue; however, PKAs have heretofore been viewed as incompatible with ad hoc and sensor networks for two reasons:

1) PKAs are much more computationally complex than symmetric key algorithms.
2) PKAs are tailored for unicast, yet in many operational scenarios, wireless networks are dominated by multicast and other forms of group communications [2]–[4].

Recent advances in lattice-based cryptography paved the way for the development of lightweight PKAs that are suitable for use in power-constrained devices [5]; the second issue, however, has received considerably less attention in the literature.

Public key algorithms can naïvely support multicast traffic by replacing each t-destination multicast session with t parallel unicast sessions. However, this approach fails to capture the energy savings afforded by, for example, multicast tree routing [6]. It is more energy efficient to have the destination clients first securely establish a group key and then derive a common public/private key pair from that group key. This allows data to be encrypted by the source, efficiently multicast to all destinations, and then decrypted by all destinations simultaneously.

A multitude of group key agreement protocols have been proposed in the literature (see, for example, [7]–[9] and Chapter 6 of [10]). The majority of these protocols extend traditional two-party Diffie-Hellman (DH) key exchange [11] to multiple parties and therefore provide a semantic security guarantee (i.e., the security depends on the intractability of the Decisional DH problem). Burmester and Desmedt's protocol (BD) [12] is particularly germane to our work as it employs multicasting and is therefore a natural fit for wireless networks. In the BD protocol, an X-bit group key is agreed upon by t clients using $2t$ public multicast transmissions, each of which is approximately X bits long. This linear growth in the number of multicasts as a function of the group size is characteristic of many existing protocols that do not rely on master keys.

A. Our Contributions

In this work, we assume that master keys are loaded on clients prior to group key agreement. Master key loading may

Manuscript received December 8, 2014; revised March 10, 2015; accepted May 11, 2015. Date of publication June 1, 2015; date of current version October 8, 2015. This work appeared in part at the 2013 IEEE Communications and Network Security Conference [1]. This research was supported by the US Defense Advanced Research Project Agency under contracts W15P7T-12-C-5013 and W911QX-13-C-0010. T. Courtade was supported in part by the NSF Center for Science of Information under grant agreement CCF-0939370. The associate editor coordinating the review of this paper and approving it for publication was K. Zeng.

T. R. Halford was with TrellisWare Technologies, Inc., San Diego, CA 92127 USA. He is now with WPL Inc., Manhattan Beach, CA 90266 USA (e-mail: tom.halford@wpli.net).

T. A. Courtade was with Stanford University, Stanford, CA 94305 USA, and also with NSF Center for Science of Information, West Lafayette, IN 47907-2066 USA. He is now with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720-1500 USA (e-mail: courtade@eecs.berkeley.edu).

K. M. Chugg is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2560 USA, and also with TrellisWare Technologies, Inc., San Diego, CA 92127 USA (e-mail: chugg@usc.edu).

X. Li and G. Thatte are with TrellisWare Technologies, Inc., San Diego, CA 92127 USA (e-mail: xli@trellisware.com; gthatte@trellisware.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2015.2439675

occur either before the network is deployed (i.e., preloading) or dynamically via Diffie-Hellman key exchanges. When a group of clients wish to establish a common key, they first derive session keys from the subset of master keys that are shared by at least two group members. The session key shared by the largest number of group members becomes the group key. That key is distributed to the remaining group members via public multicast transmissions comprising the binary sum of the group key and other session keys—i.e., shared session keys are used as one-time pads for the distribution of the group key. By properly designing the master key distribution, the number of transmissions required for group key agreement can be made to be much smaller than the group size. For example, if the master keys are distributed randomly such that a given client possesses a given key with probability $1 - 1/b$, then the number of transmissions grows with the group size t as[1] $O(\log_b t)$. This sublinear growth compares favorably to the linear growth exhibited by many existing protocols.

Owing to the use of shared session keys as one-time pads, the group key generated by our protocol is secure against out-of-network eavesdroppers in the information-theoretic sense—i.e., an eavesdropper that observes all of the public transmissions can do no better than randomly guessing the group key. However, the key may be exposed to a malicious in-network client that shares a master key from which one of the one-time pad session keys was derived. Since master keys can be revoked whenever a compromised client is detected (cf., [13]), the group key is vulnerable only to *undetected* compromised clients in practice. This vulnerability, which is common to all protocols that use master keys, may be a reasonable price to pay for increasing the energy efficiency of group key agreement in many operational scenarios.

Our approach is inspired by recent information-theoretic results on group key agreement. Building on [14], Courtade and Halford recently characterized the minimum number of public transmissions required for key agreement assuming an arbitrary master key distribution [15]. In particular, it was shown in [15] that defining a key agreement protocol that minimizes the number of transmissions is NP-hard. Our protocol employs a greedy heuristic to approximate this optimum in polynomial time. The principal results of this paper are:

1) The specification of a protocol for group key agreement that can be computed in polynomial time (in the group size t) for any distribution of the master keys.
2) For any master key distribution, the number of public transmissions required by our protocol is at most $1+H(t-1)$ times the optimum, where $H(k)$ denotes the k^{th} harmonic number. This $O(\log t)$ approximation ratio is the best possible for a polynomial time computable algorithm unless NP contains slightly superpolynomial time algorithms.
3) If $\omega(\log t)$ master keys[2] are independently allocated to each client with probability $1-1/b$, our protocol requires $O(\log_b t)$ public transmissions to generate a group key.

[1]We say that a function $f(n) = O(g(n))$ if there exists constants n_0 and c such that $f(n) \leq cg(n)$ for all values of $n > n_0$.

[2]We say $f(n) = \omega(g(n))$ if there exists n_0 such that $|f(n)| \geq c|g(n)|$ for all $n > n_0$ and for every fixed positive number c.

B. Related Work

Following Burmester and Desmedt's foundational work [12], a rich literature on group key agreement protocols has emerged (see [16] for a recent survey). Of particular relevance to our work is an extension of the BD protocol proposed by Jung [17]. In Jung's scheme, a group key is established among a set of t clients $g_1, \ldots, g_t$ by first establishing master keys between clients g_i and g_{i+1} for $i \in [1, t]$ (g_t establishes a key with g_1 to complete the cycle). Conditioned on this *cyclic* master key distribution, our protocol requires $t - 1$ public multicast transmissions. Jung's protocol employs a suboptimal transmission scheme that requires t public multicasts.

Also related to the present work are information-theoretic results on secret key generation. Characterizing the amount of communication required to generate a secret key under different models of shared randomness is a long standing problem [18]. In [19], Chan gave a suboptimal bound on the communication required for key generation under a finite linear source model that is essentially equivalent to the master key distribution model that we consider in this work. In [15], Courtade and Halford provided a complete characterization of linear[3] protocols under this model. Contemporaneous to that work, Mukherjee and Kashyap characterized secret key generation under a model in which each pair of clients shares a random string of bits that is independent of the string shared by every other pair [20]. This pairwise independent network (PIN) model, which was previously studied in [21]–[24], is more restrictive than our master key distribution model as it only allows randomness to be shared by pairs of nodes.

The primary difference between our protocol and the schemes developed to constructively prove results in [19]–[22] is its generality—i.e., we do not require a specific master key distribution.

C. Organization

We first introduce our protocol via example in Section II. Following some mathematical preliminaries in Section III, we establish our main results in Section IV, with the proof of the second main result appearing in an Appendix. In Section V we explore a number of extensions of our protocol including support for dynamic master key loading and group join operations. It is shown that over the time, the energy savings afforded by our protocol—in terms of the number of transmissions required for group key agreement—outstrip the overhead costs of dynamic master key loading. Since energy efficiency is a function of both computation and communication in power-constrained devices, we demonstrate that the computational burden of our protocol compares favorably to multi-party generalizations of Diffie-Hellman key exchange in Section VI. We conclude with directions for future work in Section VII.

II. Previewing the Protocol via Example

Consider the simple 11-client, fully-connected network illustrated in Fig. 1. A total of 21 master keys have been preloaded

[3]In a linear protocol, every public transmission is a linear combination of a subset of the master keys (or session keys derived from the master keys).

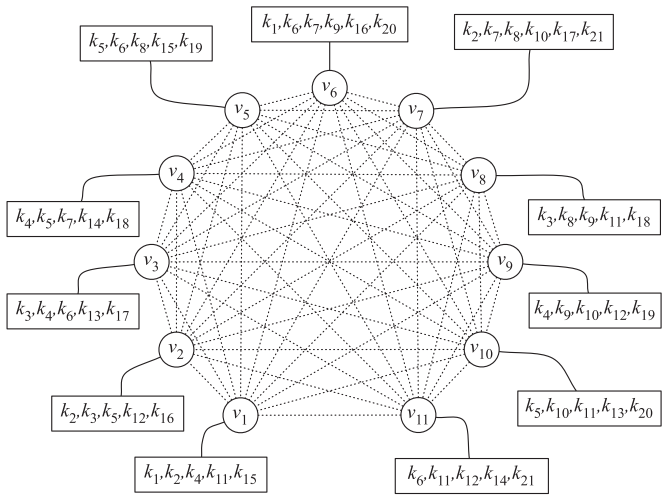

Fig. 1. A total of 21 master keys are preloaded on the 11 clients in this simple, fully-connected network.

on the clients. For example, client v_1 has been preloaded with five keys: k_1, k_2, k_4, k_{11}, and k_{15}. Each pair of clients shares at least one master key initially. For example, v_1 and v_6 share k_1. As shown in Section III, this condition allows key agreement among *any* subset of the 11 clients.

Suppose that clients v_2, v_4, v_8, v_{10}, and v_{11} wish to establish a group key for a session with unique identifier u. Observe that clients v_2, v_4, and v_{10} share master key k_5 while clients v_8, v_{10}, and v_{11} share master key k_{11}. The desired group key can be established as follows:

- Clients v_2, v_4, and v_{10} apply a common pseudorandom function[4] (PRF) to the shared master key k_5 to obtain the session key $s_{5,u} = \phi(k_5, u)$. This will be the group key.
- Clients v_8, v_{10}, and v_{11} apply a common PRF to k_{11} to obtain the session key $s_{11,u} = \phi(k_{11}, u)$. Fig. 2(a) shows the distribution of the session keys after this step.
- Client v_{10} now possesses both session keys. As illustrated in Fig. 2(b), it next transmits the bitwise sum $m_{1,u} = s_{5,u} \oplus s_{11,u}$ to clients v_8 and v_{11}, which recover the group key by computing $m_{1,u} \oplus s_{11,u} = s_{5,u}$.

Observe that in this simple example a group key has been generated among five clients via a single public multicast transmission. By using the session key $s_{11,u}$ as a one-time pad, the group key $s_{5,u}$ is secure from any eavesdropper that does not possess k_5 or k_{11} (since k_5 and k_{11} are required to generate $s_{5,u}$ and $s_{11,u}$, respectively). Furthermore, by using the output of a pseudorandom function as a group key, the security of the master keys has not been reduced—i.e., no clients obtain any new master keys.

Before describing the tools required to generalize the above example, it is instructive to consider a second example where two public multicast transmissions are required for group key agreement among a different 5-client group in the same network. Suppose that clients v_1, v_5, v_6, v_9, and v_{11} wish to establish a group key for a session with unique identifier w. The desired group key can be established as follows:

- Clients v_5, v_6, and v_{11} apply a common PRF to k_6 to obtain the group key $s_{6,w} = \phi(k_6, w)$.
- Clients v_1 and v_6 compute $s_{1,w} = \phi(k_1, w)$, while clients v_1 and v_9 compute $s_{4,w} = \phi(k_4, w)$.
- Client v_6 computes and transmits the message $m_{1,w} = s_{6,w} \oplus s_{1,w}$ to client v_1, which in turn recovers the group key via $m_{1,w} \oplus s_{1,w} = s_{6,w}$.
- Client v_1 computes and transmits the message $m_{2,w} = s_{6,w} \oplus s_{4,w}$ to client v_9, which in turn recovers the group key via $m_{2,w} \oplus s_{4,w} = s_{6,w}$.

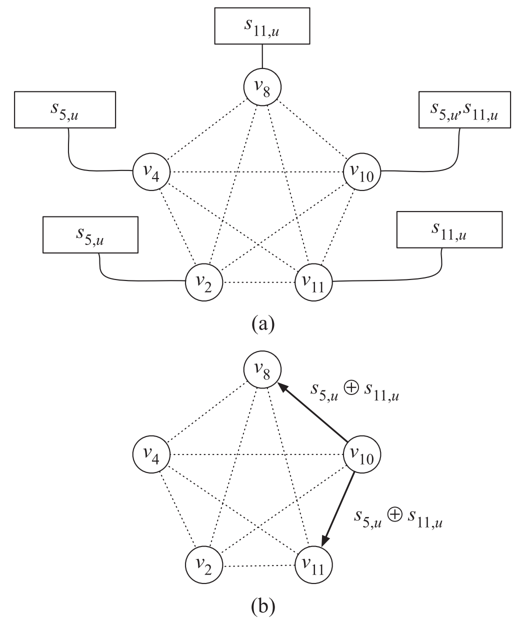

Fig. 2. After the sessions keys are derived from the master keys k_5 and k_{11} by the group members (a), client v_{10} transmits their binary sum (b) so that clients v_8 and v_{11} recover the group key $s_{5,u}$.

By using the session keys $s_{1,w}$ and $s_{4,w}$ as one-time pads, the group key $s_{6,w}$ is secure in the information-theoretic sense from any eavesdropper that possesses neither k_1 nor k_4.

Observe that in both of the above examples, other clients in the network could potentially derive the group key from the public transmissions. For example, clients v_5 and v_1 possess master keys k_5 and k_{11}, respectively, and could therefore recover the first group key. If these clients were compromised, then that key would be revealed to the attacker. This indicates that our protocol should be considered for use in ad hoc and sensor networks that also employ protocols for client compromise detection (e.g., [13] and the references therein).

III. Mathematical Building Blocks

A. Secrecy via Coded Cooperative Data Exchange

Our approach is motivated by recent results on secret key agreement via the coded cooperative data exchange (CCDE) problem. Introduced first by El Rouayeb *et al.* in [26], the CCDE problem has received significant attention (e.g., [14], [27]) and

[4]A PRF family is a set of polynomial time computable functions $\{\phi(x, s) | s \in S\}$ of some input variable x that are indexed by a seed parameter s such that when s is selected randomly from the set of possible seeds S, $\phi(x, s)$ is computationally indistinguishable from a random function [25]. In practice, a keyed-hash message authentication code (HMAC) could be used.

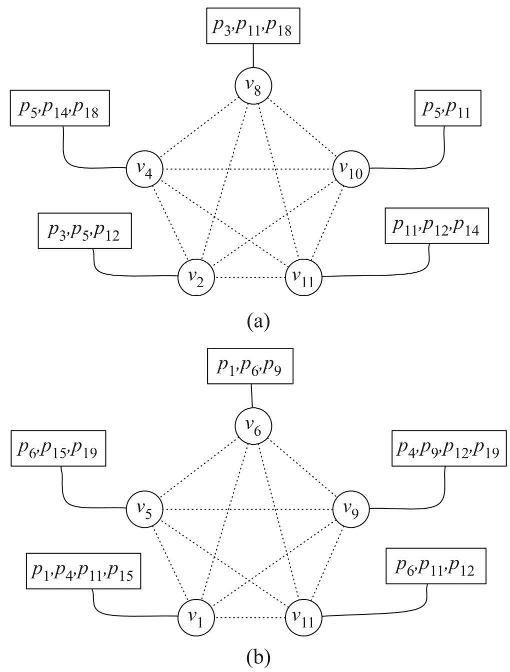

Fig. 3. Definition of the coded cooperative data exchange problem among two 5-client groups within the network of Fig. 1. A total of 4 (resp., 6) transmissions are required for the optimal solution to the CCDE problem defined by (a) [resp., (b)]. Both solutions generate 2 packets of secrecy.

is stated as follows. Suppose that k packets are distributed among a network of t clients. What is the minimum number of transmissions M required to recover all k packets at all t clients? In [28], it was shown that if the network[5] is fully connected, then the CCDE problem can be solved in polynomial time. At roughly the same time, Chan [19] and Milosavljevic *et al.* [32] independently established similar results. In general, less than k transmissions are required by optimal CCDE solutions in fully-connected networks. The difference betweeen k and M can be exploited for secret key agreement.

In Fig. 3(a) and (b), we revisit the groups studied in the two examples of Section II. The master keys $\{k_i\}_{i=1}^{21}$ have been replaced by packets $\{p_i\}_{i=1}^{21}$ with the same indices for consistency with the notation used in the CCDE literature. Observe that only those packets shared by at least two clients are listed in these figures. This is because any packet possessed by only one of the clients in a group does not contribute to the CCDE solution in an interesting way (i.e., that packet must simply be broadcasted to the other group members so that the difference between k and M does not change).

A total of $k = 6$ packets have been distributed among the clients in Fig. 3(a). Using the tools of [14], it can be shown that these $M = 4$ transmissions define an optimal CCDE solution:

(i) v_2 transmits $m_1 = p_3 \oplus p_5$. v_4 and v_{10} recover $p_3 = m_1 \oplus p_5$. v_8 recovers $p_5 = m_1 \oplus p_3$.

(ii) v_4 tranmits $m_2 = p_5 \oplus p_{14}$. v_2, v_8, and v_{10} all recover $p_{14} = m_2 \oplus p_5$. v_{11} recovers $p_5 = m_2 \oplus p_{14}$ and then $p_3 = m_1 \oplus m_2 \oplus p_{14}$.

(iii) v_8 transmits $m_3 = p_{11} \oplus p_{18}$. v_4 recovers $p_{11} = m_3 \oplus p_{18}$. v_{10} and v_{11} recover $p_{18} = m_3 \oplus p_{11}$.

(iv) v_{11} transmits $m_4 = p_{12} \oplus p_{18}$. v_4, v_8, and v_{10} all recover $p_{12} = m_4 \oplus p_{18}$. Finally, v_2 recovers $p_{18} = m_4 \oplus p_{12}$ and then $p_{11} = m_3 \oplus m_4 \oplus p_{12}$.

The results of [18] indicate that if all 6 packets are cryptographic keys, then this scheme generates precisely $k - M = 2$ packets worth of secrecy. In this example, p_5 and p_{18} can form the secret.

Similarly, $k = 8$ packets are distributed among the clients in Fig. 3(b). It can be verified that the following $M = 6$ transmissions comprise a CCDE solution with p_1 and p_{15} as the secret: (i) v_1 transmits $p_4 \oplus p_{15}$, (ii) v_5 transmits $p_6 \oplus p_{15}$, (iii) v_6 transmits $p_1 \oplus p_9$, (iv) v_9 transmits $p_1 \oplus p_{12}$, (v) v_9 transmits $p_1 \oplus p_{19}$, and (vi) v_{11} transmits $p_{11} \oplus p_{12}$.

In these examples, two packets worth of secrecy were generated using 4 (resp., 6) public multicast transmissions that resulted in the clients recovering 6 (resp., 8) packets. Since group key agreement requires only a *single* secret packet, a full CCDE solution may not be necessary for our application. Indeed, in the first example of Section II, one secret packet was generated using one public multicast transmission and only two packets were recovered by the destination clients ($s_{5,u}$ and $s_{11,u}$). This observation motivates the following definition.

Definition 1: Let a set of k packets $p_1, \dots, p_k$ be distributed among n clients $v_1, \dots, v_n$. A **group key agreement protocol** for that packet distribution is specified by $m \leq n$ encoding functions $f_{i_1}(\cdot), \dots, f_{i_m}(\cdot)$ and n decoding functions $g_1(\cdot), \dots, g_n(\cdot)$ such that:

P-1. For each $j \in [1, m]$, the inputs to the encoding function $f_{i_j}(\cdot)$ depend only on the packets possessed initially by client v_{i_j}.

P-2. For each $i \in [1, n]$, the inputs to the decoding function $g_i(\cdot)$ depend on the packets possessed initially by client v_i and on the output of the encoding functions $f_{i_j}(\cdot)$ for all $j \in [1, m]$.

P-3. The output of every decoding function is a common packet x. This packet is the group key.

P-4. There is zero mutual information between x and the outputs of the encoding functions $f_{i_1}(\cdot), \dots, f_{i_m}(\cdot)$.

If for all $j \in [1, m]$, client v_{i_j} evaluates and transmits $f_{i_j}(\cdot)$, then all of the clients can recover x by evaluating their respective decoding functions. Property P-4 ensures that the common packet x that is recovered by all of the clients is a secret key.

Observe that in the first example of Section II, we implicitly defined a group key agreement protocol in which all of the clients recover the session key $s_{5,u}$. There is a single encoding function in that example corresponding to the message transmitted by client v_{10},

$$f_{10}(s_{5,u}, s_{11,u}) \triangleq s_{5,u} \oplus s_{11,u}, \tag{1}$$

and the clients employ one of two decoding functions depending on whether they initially possess $s_{5,u}$ or if they must obtain

[5]While the networks considered in this work need not be fully connected at the physical layer (PHY), cryptographic protocols are often implemented at higher layers that see an abstracted, one-hop topology. Moreover, a number of emerging wireless stacks employ cooperative flooding protocols at the PHY and present a fully-connected topology to higher layers [29]–[31].

it from the message transmitted by client v_{10}:

$$g_2(s_{5,u}) = g_4(s_{5,u}) = g_{10}(s_{5,u}) \triangleq s_{5,u},$$
$$g_8\big(s_{11,u}, f_{10}(\cdot)\big) = g_{11}\big(s_{11,u}, f_{10}(\cdot)\big) \triangleq s_{11,u} \oplus f_{10}(\cdot) = s_{5,u}. \quad (2)$$

Owing to the security of the XOR operator, there will be zero mutual information between the transmission $f_{10}(\cdot)$ and the group key $s_{5,u}$ provided that $s_{5,u}$ and $s_{11,u}$ are secret keys.

In this work we seek energy-efficient group key agreement protocols—i.e., those requiring as small a number of public multicast transmissions as possible. This motivates the following definition.

Definition 2: Let a set of k packets $p_1, \ldots, p_k$ be distributed among n clients $v_1, \ldots, v_n$. A group key agreement protocol is said to be **optimal** if it requires the fewest number of transmissions possible over all group key agreement protocols for that packet distribution.

Optimal group key agreements protocols need not be unique. For example, in the packet distribution illustrated in Fig. 3(b), 15 different optimal group key agreement protocols can be identified that establish p_6 as a secret with two transmissions.

B. Key Agreement via Connected Spanning Subhypergraphs

Suppose that k packets $P = \{p_j\}_{j \in J}$ are distributed among n clients $V = \{v_i\}_{i \in I}$. For each packet index $j \in J$, let $e_j \subseteq V$ be the subset of clients in possession of packet p_j. This packet distribution can be described graphically per Figs. 1–3 or, equivalently, by the hypergraph $H(V, E_H)$ with vertex set V and hyperedge set $E_H = \{e_j\}_{j \in J}$.

A hypergraph $H(V, E_H)$ is said to be *connected* if for every non-empty proper subset of the vertex set $U \subset V$, there exists a hyperedge incident on some vertex in U and on another vertex in $V \setminus U$. The following result, which was first presented as part of [15, Lemma 4], provides a necessary and sufficient condition for the existence of a group key agreement protocol for a given packet distribution.

Lemma 1: Let $H(V, E_H)$ be the hypergraph implied by a distribution of k packets P among n clients V. A group key agreement protocol can be defined for this packet distribution if and only if $H(V, E_H)$ is connected.

Proof: Suppose that $H(V, E_H)$ is connected. We construct a group key agreement protocol as follows. Select any hyperedge $e_s \in E_H$ and set $U = e_s$. The packet p_s corresponding to e_s can be recovered by all clients by repeating the following steps until $U = V$:

1) Select a hyperedge $e_j \in E_H$ that is incident on a vertex $v_i \in U$ and at least one in $V \setminus U$.
2) v_i transmits the binary sum $m_j = p_s \oplus p_j$, where p_j corresponds to hyperedge e_j.
3) All clients in e_j recover $p_s = m_j \oplus p_j$.
4) Update U to include all clients now possessing p_s—i.e., $U \leftarrow U \cup e_j$.

Since $H(V, E_H)$ is connected, a hyperedge can be found in Step 1 as long as $U \neq V$. Since all of the transmissions are of the form $p_s \oplus p_t$, there will be zero mutual information between the group key p_s and any of the transmissions, provided all packets are secret keys.

To prove the converse, suppose that $H(V, E_H)$ is not connected. By definition, there exists some non-empty subset of the clients $U \subset V$ such that there are no hyperedges connecting vertices in U to those in $W = V \setminus U$. That is to say, there are no packets that are shared by a client in U and one in W. It follows from Theorem 6 of [14] that precisely zero packets of secret key can be generated via a solution to the CCDE problem with such a packet distribution. Therefore, a group key agreement protocol cannot be defined for this packet distribution. $\square$

Lemma 1 identifies a necessary and sufficient condition for secret key generation given some arbitrary master key distribution. The following result follows immediately from Lemma 1 and provides guidance on how to design master key distributions.

Proposition 1: A group key agreement protocol can be defined among any subset of the clients in a network if and only if every pair of clients share at least one master key.

In the proof of Lemma 1, we specified a group key agreement protocol by identifying a subset of hyperedges $\widetilde{E}_H \subseteq E_H$ that spans the vertex set V and which induces a connected subhypergraph. Each hyperedge $e_j \in \widetilde{E}_H$ corresponds to an encoding function and a transmission. This suggests that to define an energy-efficient group key agreement protocol, we should search for connected subhypergraphs of $H(V, E_H)$ that span V with the fewest possible hyperedges. Indeed, [15, Lemma 4] implies that optimal group key agreement protocols coincide with solutions to the Minimum Connected Subhypergraph (MCSH) problem on $H(V, E_H)$. As shown in [15, Theorem 4], a connection between the MCSH problem and the NP-complete Set Cover problem can be used to show that defining an optimal group key agreement protocol is NP-hard. Fortunately, as with many problems related to Set Cover, the MCSH problem can be approximated in polynomial time using a greedy heuristic. As will be shown in Section IV, a greedy approximation of the MCSH problem forms the basis of our protocol.

IV. Group Key Agreement with Preloaded Master Keys

A. Protocol Specification

We now describe our protocol for group key agreement under the assumptions that (i) the clients have been loaded with master keys according to a distribution satisfying Proposition 1 and (ii) the network is fully connected. In Section V we will consider extensions that support dynamic master key exchange and extensions that account for multi-hop network topologies.

Let $G = \{g_1, \ldots, g_t\}$ be a set of t clients that wish to agree upon a group key for a session with unique identifier u. Client g_i possesses the set of master keys with indices tabulated by K_i. We define the occupancy set $O_j \subseteq G$ for each master key index $j \in K_G = \cup_{i=1}^{t} K_i$ as the subset of G that possess the key k_j—i.e., $g_i \in O_j$ if and only if $j \in K_i$. Provided that the occupancy sets $\mathcal{O} = \{O_j\}_{j \in K_G}$ are known to all group members, then Algorithm 1 can be run in parallel at each client to establish the desired group key. Algorithm 1 determines the transmissions used for group key agreement by applying a simple greedy heuristic to the minimal connected spanning subhypergraph problem implied by the distribution of the master keys indexed by K_G (hyperedges) on the clients in G (vertices).

Algorithm 1 Proposed protocol for group key agreement running at client $g_i \in G$.

Input: Occupancy sets $\mathcal{O} = \{O_j\}_{j \in K_G}$, group $G = \{g_1, \ldots, g_t\}$, and a common PRF $\phi()$.
Output: Group key $s_{j_0,u}$ for session with unique identifier u.

$j_0 \leftarrow$ index of largest occupancy set in $\mathcal{O}$, $C \leftarrow O_{j_0}$, $l \leftarrow 1$;
if $g_i \in O_{j_0}$ **then**
  compute the group key $s_{j_0,u} \leftarrow \phi(k_{j_0}, u)$;
end
while $C \neq G$ **do**
  $j_l \leftarrow$ index of an occupancy set $O_{j_l} \in \mathcal{O}$ satisfying $O_{j_l} \cap C \neq \emptyset$ that maximizes $|O_{j_l} \setminus C|$;
  $i_l \leftarrow$ a client in $O_{j_l} \cap C$;
  if $g_i = i_l$ **then**
    compute the l^{th} one-time pad $s_{j_l,u} \leftarrow \phi(k_{j_l}, u)$;
    compute the bit-wise sum $m_{l,u} = s_{j_0,u} \oplus s_{j_l,u}$;
    transmit $m_{l,u}$ to all clients in $O_{j_l} \setminus C$;
  else if $g_i \in O_{j_l} \setminus C$ **then**
    compute the l^{th} one-time pad $s_{j_l,u} \leftarrow \phi(k_{j_l}, u)$;
    receive $m_{l,u}$ from client i_l;
    recover the group key $s_{j_0,u} = m_{l,u} \oplus s_{j_l,u}$;
  end
  $C \leftarrow C \cup O_{j_l}$, $l \leftarrow l + 1$;
end

To clarify the notation used in Algorithm 1, we revisit the second example of Section II wherein the $t = 5$ clients $G = \{v_1, v_5, v_6, v_9, v_{11}\}$ wish to establish a group key for a session with unique identifier w. In this example, the occupancy sets with at least two elements are:

$$\begin{aligned} O_1 &= \{v_1, v_6\}, O_4 = \{v_1, v_9\}, O_6 = \{v_5, v_6, v_{11}\}, \\ O_9 &= \{v_6, v_9\}, O_{11} = \{v_1, v_{11}\}, O_{12} = \{v_9, v_{11}\}, \\ O_{15} &= \{v_1, v_5\}, O_{19} = \{v_5, v_9\}. \end{aligned} \tag{3}$$

The largest occupancy set is O_6 so Algorithm 1 begins by setting $j_0 = 6$, $C = O_6$, and $l = 1$. Clients v_5, v_6, and v_{11} next compute the group key $s_{6,w} = \phi(k_6, w)$. In the first iteration of the while loop, there are 6 occupancy sets that contain precisely one element in C and one element not in C. To break the tie, the occupancy set with the lowest master key index is chosen so that $l_1 = 1$ and $i_1 = 6$. Client v_6 thus computes $s_{1,w} = \phi(k_1, w)$ and transmits the binary sum $m_{1,w} = s_{6,w} \oplus s_{1,w}$. Client v_1 subsequently computes $s_{1,w}$, receives $m_{1,w}$, and recovers the group key $s_{6,w} = m_{1,w} \oplus s_{1,w}$. The first iteration concludes by setting $C = \{v_1, v_5, v_6, v_{11}\}$ and $l = 2$. The second iteration proceeds is a similar manner with $l_2 = 4$ and $i_2 = 1$. After two iterations, $C = G$ and the group key has been recovered by all 5 clients.

B. Protocol Discussion

The common PRF that is employed in Algorithm 1 ensures that the public transmissions are computationally indistinguishable from random packets, thereby establishing our first key result.

Result 1: Algorithm 1 specifies a group key agreement protocol for any set of clients in a network that has been loaded with master keys according to a distribution that satisfies Proposition 1.

Recall that specifying an optimal group key agreement protocol is NP-hard. Nevertheless, the polynomial time greedy heuristic employed in Algorithm 1 provides a group key agreement protocol with an $O(\log t)$ approximation ratio. Our second key result is proved in the Appendix.

Result 2: The number of transmissions required by Algorithm 1 is at most $1 + H(t-1)$ times that of an optimal group key agreement protocol, where $H(t)$ denotes the t^{th} harmonic number. This $O(\log t)$ approximation ratio is the best possible for a polynomial time computable algorithm unless NP contains slightly superpolynomial time algorithms.

Result 1 implies that the group key established by our protocol is secure against out-of-network eavesdroppers in the information-theoretic sense [18]. This is a stronger security guarantee than that provided by protocols based on Diffie-Hellman key exchange. Of course, undetected compromised clients can potentially recover the group key by eavesdropping on the transmissions used for key agreement. This is the price that we pay for group key agreement among t clients with far fewer than t transmissions. As discussed above, however, this security vulnerability can be mitigated by the use of a protocol for detecting compromised clients.

We use session keys derived from master keys in our protocol to provide forward and backward security [33]. That is to say, an adversary not possessing any of the master keys but possessing a subset of the group keys cannot discover another group key in our protocol. In practice, an HMAC could be used as the PRF with the session identifier as an input variable and the master key as the seed parameter. This approach is consistent with recommendations by the National Institute of Standards [34] for ensuring that the compromise of a session or group key does not degrade the cryptographic strength of the corresponding master key.

C. Simulation Results—Energy Efficiency

In our simulations, we consider master key loading schemes where R keys are distributed randomly[6] among n clients such that any client possesses any master key with probability β. Fig. 4 compares the average number of public multicast transmissions required for group key agreement by Algorithm 1 when $n = 50$, $R = \lceil n/\beta \rceil$, and β varies. Observe that as β increases, the number of transmissions required for key agreement decreases. This is consistent with intuition: as the occupancy set sizes increase, the minimal connected subhypergraph solution size decreases. To highlight the sublinear growth achieved by our protocol, the linear growth exhibited by Burmester and Desmedt's protocol [12] is also shown. The difference in performance between the two protocols can be attributed largely to

[6]When required by Lemma 1 for a given group, pairwise keys are added to the random distribution. Master key distributions derived from combinatorial designs are an alternative approach to providing random-like key distributions that satisfy Proposition 1. Combinatorial designs have been studied extensively in the context of the sensor network key distribution problem (see, e.g., [35]). Singer difference sets in particular are well-suited to our protocol.

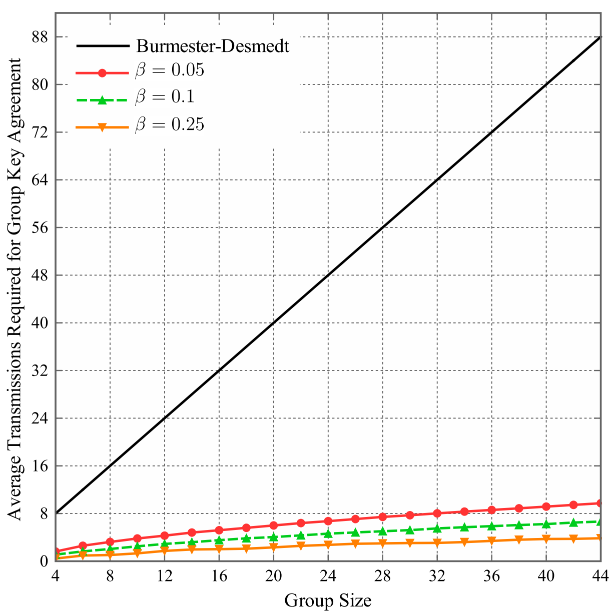

Fig. 4. Number of public multicast transmissions required for group key agreement via the proposed protocol when $\lceil 50/\beta \rceil$ master keys are loaded randomly in a 50-client network. Our protocol exhibits a sublinear growth in the number of transmissions as the group size increases.

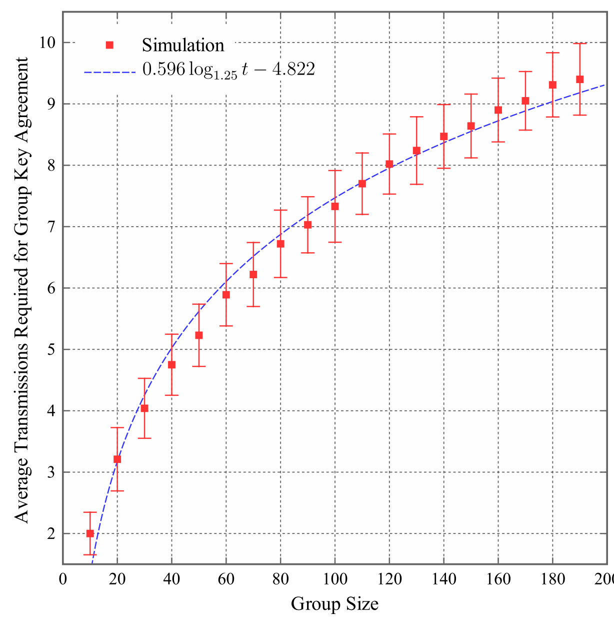

Fig. 5. Number of public multicast transmissions for group key agreement for the proposed protocol when $n = 200$, $R = 100$, $\beta = 0.2$, and $b = 1.25$. The number of transmissions required for key agreement grows as the logarithm of the group size.

our use of session keys derived from preloaded master keys—i.e., the BD protocol instead generates all keys on-the-fly.

The MCSH problem assuming a random master key distribution is closely related to random instances of the Set Cover problem. Toward connecting the two, consider the Set Cover problem with ground set $X = \{1, 2, \ldots, t\}$ and subsets $S_i \subseteq X$, $i = 1, \ldots, R$. A (β, t, R)-random instance of the Set Cover problem is generated by letting an element of the ground set $x \in X$ be a member of S_i with probability β, independently of all other elements. That is:

$$\Pr\{x \in S_i\} = \beta$$

independently for all $x \in X$, $i \in \{1, \ldots, R\}$. Building on [36], Telelis and Zissimopoulos [37] investigated greedy approximations to (β, t, R)-random instances of the Set Cover problem, and showed that with high probability, the size of a greedily chosen set cover grows as

$$O\left(-\frac{\log t}{\log(1-\beta)}\right), \quad (4)$$

provided $R = \omega(\log t)$. This constraint on the number of subsets guarantees the existence of a feasible solution with probability one.

Returning to the MCSH problem assuming a random master key distribution, suppose that instead of applying the greedy heuristic of Algorithm 1, we instead employ the following two-step approach. First, we identify a size-S subset of the master keys whose occupancy sets cover the group. This is done using the greedy Set Cover approximation algorithm studied in [37]. Second, we augment that subset with pairwise keys as necessary to ensure that the subhypergraph implied by the selected master keys is connected. Since at most $S - 1$ pairwise keys need to be added in this step, the size of the MCSH approximation grows as $O(\log_b t)$, where $b = 1/(1 - \beta)$.

Result 2 indicates that this two-step approach will not outperform the greedy heuristic of Algorithm 1. Thus, we can make a more precise statement about the apparent logarithmic growth observed in Fig. 4. Although Result 3 is only guaranteed to hold in an asymptotic sense, Fig. 5 suggests that this predicted behavior holds for finite n and t.

Result 3: If $R = \omega(\log t)$ master keys are independently allocated to each client with probability $1 - 1/b$, Algorithm 1 requires $O(\log_b t)$ public multicast transmissions to generate a group key with high probability.

D. Simulation Results—Energy vs. Security Trades

Fig. 6 illustrates the tradeoff between energy efficiency and security against undetected compromised clients in the proposed protocol. Energy efficiency is measured in terms of the number of public multicast transmissions. For different values of β, we measured the average number of public multicast transmissions required for group key agreement among 10, 15, 20, and 25 clients in a 50-client network. The total number of keys R was set to $\lceil 50/\beta \rceil$ so that the average number of keys per client remained constant. Simultaneously, we measured the average number of clients in the network that can recover the group key. This includes the desired group members as well as any other clients that posses the master keys used to generate the sessions keys in Algorithm 1. Depending on the likelihood that a compromised in-network node will go undetected, different master key loading scheme parameters should be chosen in practice. The results illustrated in Fig. 6 indicate how this parameter selection will impact the energy efficiency of group key agreement in our protocol.

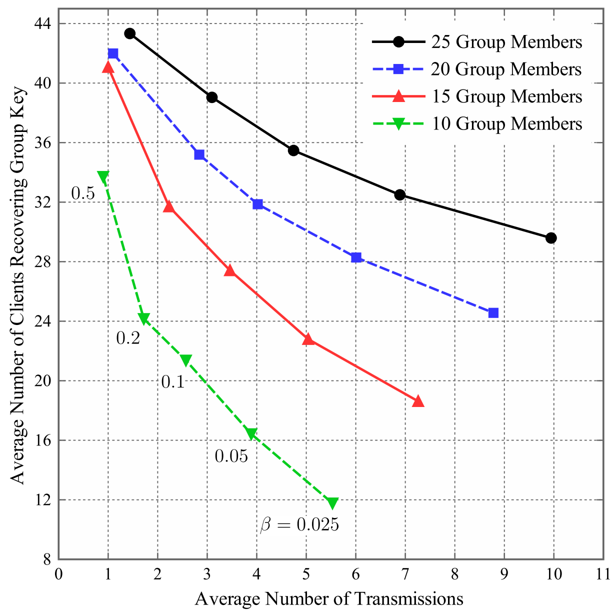

Fig. 6. The average number of clients that can recover the group key as a function of the average number of transmissions required to generate it. The tradeoffs illustrated here, along with the likelihood that a compromised node goes undetected, can inform master key loading scheme parameter selection.

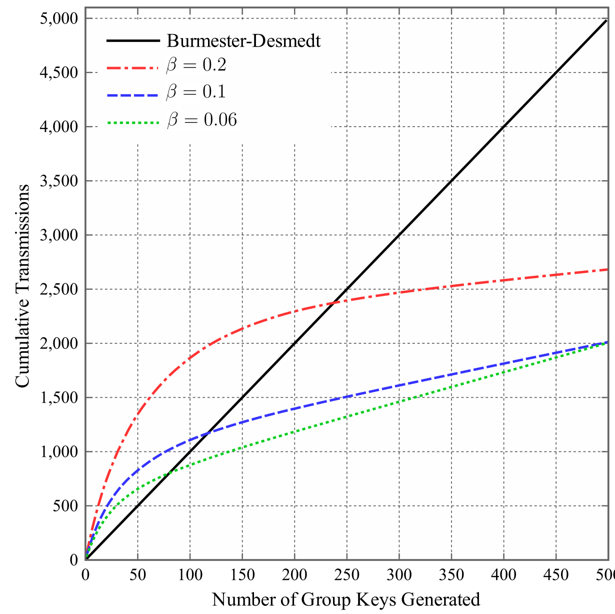

Fig. 7. Cumulative number of transmissions required to exchange master keys and establish group keys among 5-client groups that are randomly selected according to the proximity multicast model. The cost of master key exchange is amortized over time and our dynamic protocol outperforms the BD protocol.

V. Extensions to the Protocol

A. Group Key Agreement With Dynamic Master Key Exchange

The protocol described in Section IV assumes that master keys have already been loaded on the clients. In this section, we describe how our protocol can be extended to support on-the-fly master key exchange and group key agreement.

1) Protocol Description: In the dynamic master key exchange variant of our protocol, we assume that every client is fielded with the ability to generate cryptographic keys. Client v_i initially generates and stores f_i random master keys, where f_i is a binomial random variable drawn from the distribution

$$\Pr\{f_i = m\} = \binom{n}{m}\alpha^m(1-\alpha)^{n-m}, \qquad (5)$$

with $\alpha = R/n^2$ chosen so that a total of R random master keys are generated on average. These keys propagate through the network via an epidemic model that is inspired by distributed database maintenance algorithms [38]. Each client is initially *infected* by the keys it possesses and *susceptible* to all other master keys in the network. As clients interact to establish group keys, they become infected by or *immunized* to master keys possessed by other group members.

Suppose that the clients in $G=\{g_1,\ldots,g_t\}$ wish to establish a group key. Before running Algorithm 1, each pair of clients $g_i \neq g_j$ first runs the following master key exchange procedure.

- **Pairwise Key Agreement**: If clients g_i and g_j have not previously interacted, then they establish a key for secure pairwise communications via a traditional two-party Diffie-Hellman key exchange.
- **Random Key Exchange**: Let client g_i (resp., g_j) possess the random keys indexed by K_i (resp., K_j).
  - For each $l \in K_i$ to which g_j is susceptible, g_j becomes infected by random key k_l with probability β and immune to k_l with probability $1-\beta$. If g_j is infected by k_l, then k_l is securely transmitted by g_i to g_j (using the pairwise key). Conversely, if g_j becomes immunized to k_l, then it will never receive that key.
  - For each $m \in K_j$ to which g_i is susceptible, client g_i becomes infected by k_m with probability β, prompting a secure transmission of k_m from g_j to g_i. With probability $1-\beta$, client g_i instead becomes immunized to k_m and will subsequently never obtain that random key.

Note that the pairwise keys are exchanged to (i) provide a means for secure random key exchange and (ii) ensure that Lemma 1 is met by the master key distribution on G. At steady state, each client will be infected by an average of βR random master keys, and a given random master key will be incident on a given client with a probability that approaches β. Observe that our dynamic master key exchange procedure readily supports extensions for master keys with finite lifetimes—i.e., new keys could be generated as old ones expire.

2) Protocol Discussion: In order to study the dynamics of the proposed extension of our protocol, we simulated a network with 50 clients distributed randomly in a square. The t closest clients to a randomly chosen source define the destination set that establish a given group key. This *proximity* multicast model is representative of military use cases.

Fig. 7 illustrates how the cumulative number of transmissions required for master key exchange and group key agreement evolves over time in a 50-client network under the proximity multicast model. The multicast group size is fixed to $t = 5$ and the number of random keys is set to $\lceil \log_2 n/\beta \rceil$. Each time a group key is generated, the total number of transmissions required for pairwise and random key agreement is tabulated in addition to those required for group key agreement via Algorithm 1. Initially, there is a sharp increase in the cumulative number of transmissions as pairwise keys are established and random keys propagate via an epidemic model. Over time, the cost of master key exchange is amortized and the slopes of the curves in Fig. 7 converge to roughly 2 transmissions per generated group key.

Observe in Fig. 7 that increasing the infection probability from $\beta = 0.06$ to 0.2 decreases the number of transmissions required to establish group keys at steady state but increases the overhead associated with the random key exchange step. Setting $\beta = 0.1$ appears to offer a good trade between the steady-state and transient behavior.

For comparison, Fig. 7 also illustrates the cumulative number of transmissions when the BD protocol is used for group key agreement. Since $t = 5$, this is simply a line with slope $2t = 10$. Observe that after approximately 120 group keys have been generated, the proposed protocol with $\beta = 0.1$ becomes more energy-efficient than the BD protocol. That is to say, over time the energy savings afforded by each group key agreement in our protocol outstrip the overhead incurred for dynamic master key exchange. Note that about 2 multicasts are required per generated group key in our protocol versus 10 per group key for the BD protocol.

B. Topology-Aware Group Key Agreement

The protocol described in Section IV seeks to minimize the total number of multicast transmissions required for group key agreement. This is a useful proxy for energy efficiency in one-hop networks and in emerging wireless network approaches that employ cooperative flooding protocols [29]–[31]. However, in multi-hop wireless networks that employ more traditional tree-based multicast routing protocols, we should also account for the energy costs of relaying. In this section, we describe how our protocol can be extended for use in such networks.

1) Protocol Description: The depth of the tree used for multicast routing is a useful proxy for the energy-efficiency of multicast in many wireless networks [39]. Let $h(v_i, v_j)$ be the distance in hops between clients v_i and v_j and let

$$h(s, D) = \max_{d \in D} h(s, d) \tag{6}$$

be the depth of a minimum-depth multicast tree from a source s to a destination set D. Algorithm 2 extends Algorithm 1 so as to minimize the sum of the depths of the multicast trees used for group key agreement rather than the number of multicasts. Recall that in each iteration of the while loop in Algorithm 1, the occupancy set O_{j_l} that maximizes the number of new clients obtaining the group key,

$$|O_{j_l} \setminus C|, \tag{7}$$

is identified (where C is the set of clients already in possession of the group key). In each iteration of the while loop in Algorithm 2, the occupancy set O_{j_l} and transmitter $i_l \in O_{j_l} \cap C$ that maximizes the number of new clients obtaining the group key *per hop*,

$$\frac{|O_{j_l} \cap C|}{h(i_l, O_{j_l} \cap C)}, \tag{8}$$

is instead identified. This topology-aware heuristic extends the standard approximation algorithm for weighted set cover [40] to the hypergraph setting.

Algorithm 2 Topologically-aware protocol for group key agreement running at client $g_i \in G$.

Input: Occupancy sets $\mathcal{O} = \{O_j\}_{j \in K_G}$, group $G = \{g_1, \ldots, g_t\}$, hop distance $h(g_i, g_j)$ between all pairs of clients, and a common PRF $\phi()$.
Output: Group key $s_{j_0,u}$ for session with unique identifier u.

```
j_0 ← index of largest occupancy set in O;
C ← O_{j_0}, l ← 1;
if g_i ∈ O_{j_0} then
    compute the group key s_{j_0,u} ← φ(k_{j_0}, u);
end
while C ≠ G do
    (i_l, j_l) ← index of an occupancy set O_{j_l} ∈ O
            satisfying O_{j_l} ∩ C ≠ ∅ and a transmitter
            i_l ∈ O_{j_l} ∩ C that maximizes the number of
            new clients that will obtain s_{j_0,u} per hop:
            |O_{j_l} ∩ C| / h(i_l, O_{j_l} ∩ C);
    if g_i = i_l then
        compute the l^th one-time pad s_{j_l,u} ← φ(k_{j_l}, u);
        compute the bit-wise sum m_{l,u} = s_{j_0,u} ⊕ s_{j_l,u};
        multicast m_{l,u} to all clients in O_{j_l} \ C;
    else if g_i ∈ O_{j_l} \ C then
        compute the l^th one-time pad s_{j_l,u} ← φ(k_{j_l}, u);
        receive m_{l,u} from client i_l;
        recover the group key s_{j_0,u} = m_{l,u} ⊕ s_{j_l,u};
    end
    C ← C ∪ O_{j_l}, l ← l + 1;
end
```

2) Protocol Discussion: We compared the performance of the protocols described by Algorithms 1 and 2 in a $n = 100$ client network using a random master key distribution with $R = 100$ and $\beta = 0.2$. To induce a random geometric graph topology, the clients were placed randomly in a unit square and a transmission radius of

$$r(n) = \frac{3}{2}\sqrt{\frac{\log n}{\pi n}} \tag{9}$$

was assumed [41]. Multicast groups were selected randomly rather than via the proximity model considered in Section V-A. Fig. 8 compares the average cost of group key agreement in the two protocols as measured by two proxies for energy efficiency: number of multicasts and the sum of the multicast tree depths. Although the topology-aware protocol requires more multicasts, the sum of the tree depths over those multicasts is nearly halved with respect to the topology-agnostic protocol. Changing the optimization criteria also had security implications in this experiment. Fig. 9 compares the average number of clients that can obtain the group key under two protocols under

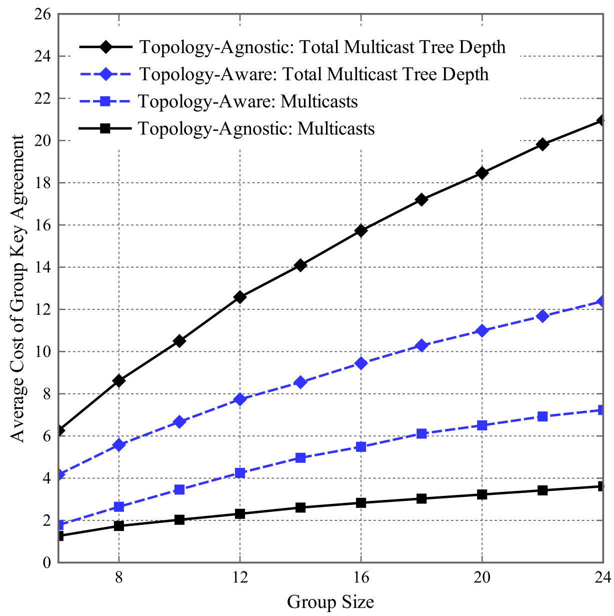

Fig. 8. Average cost of group key agreement of the topology-agnostic (Algorithm 1) and topology-aware (Algorithm 2) variants of our protocol in an $n = 100$ client network.

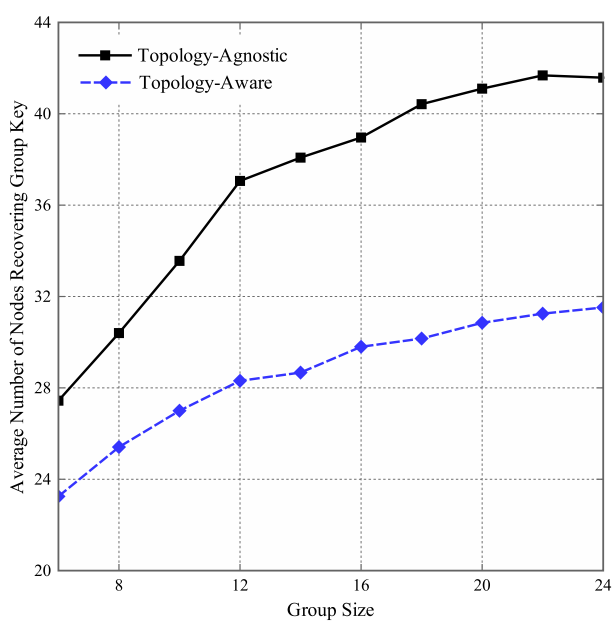

Fig. 9. Average number of clients that can recover the group key as a function of the group size in the topology-agnostic and topology-aware protocols. By constraining the tree-depth of each multicast transmission, fewer unintended clients can recover the group key under the topology-aware variant.

the assumption that all clients within h hops of the source of a depth-h multicast transmission can eavesdrop on that transmission, but clients outside of that hop radius cannot. The results shown in Fig. 9 are consistent with intuition: if the depth of every multicast tree is minimized, then so will be the number of unintended receivers.

C. Group Join Operations

In [7], Steiner *et al.* defined a family of auxiliary key operations required for dynamic group support—e.g., group member join and leave—and proposed a multi-party extension of Diffie-Hellman key exchange that supports these operations. The protocols presented in this paper can be readily extended to support group member join operations. Indeed, the mass join of j new group members to an existing t-client group can be achieved with fewer than j transmissions by suitably adapting Algorithm 1. However, member leave operations are not readily supported in our protocols. When a member leaves, the group key must be refreshed.

D. Network Join Operations

Our protocols assume global knowledge of the master key distribution. That distribution must therefore be communicated to any new client joining the network. Suppose that R keys are distributed among n clients such that any client possesses any master key with probability β. This random master key distribution can be described by an $n \times R$ incidence matrix or by a list of the keys possessed by each client. Since each client possesses βR master keys on average,

$$M(n, \beta, R) = \min\left(nR, n\beta R\lceil\log_2 R\rceil\right) \tag{10}$$

bits are required to describe the random master key distribution. For the networks considered in Figs. 4 and 5, respectively, $M(50, 0.1, 500) = 22\,500$ bits and $M(200, 0.2, 100) = 20\,000$ bits. The superlinear growth of $M(n, \beta, R)$ with n for the random master key distribution may not be be satisfactory in practice. Distributions derived from combinatorial designs or pseudo-random distributions derived from a single random seed could instead be used to control the overhead associated with network join operations. Since the security of our protocol does not depend on the eavesdropper's knowledge of the master key distribution, such distributions are permitted.

VI. Computational Complexity

In Section IV, it was shown that the protocol defined in Algorithm 1 compares favorably to Burmester and Desmedt's group key agreement protocol in terms of the number of transmissions. While communication is typically the most significant factor in energy consumption in wireless sensor networks [42], computation is also important in power-constrained devices. In this section, we show that our protocol also compares favorably to the BD protocol in terms of computation.

A. Review of Burmester and Desmedt's Group Key Agreement Protocol

Let p be a cX-bit prime for some $c \geq 1$ and let $\alpha \in \mathbb{Z}_p$ have order q, where q is an X-bit number. Suppose that a set of t clients $G = \{g_1, \ldots, g_t\}$ wish to establish an X-bit secure group key. In [12], Burmester and Desmedt described the following protocol for group key agreement:

1) Each client g_i randomly generates an integer modulo q, r_i, and broadcasts $z_i = \alpha^{r_i} \bmod p$.

2) Upon reception of z_j for all $1 \le j \ne i \le t$, each client g_i computes and broadcasts

$$X_i = \left(\frac{z_{i+1}}{z_{i-1}}\right)^{r_i} \mod p,$$

where the indices are taken in a cycle so that $z_{t+1} = z_1$ and $z_0 = z_t$.

3) Upon reception of X_j for all $1 \le j \ne i \le t$, each client g_i computes the group key

$$K = (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i-2} = \alpha^{r_1 r_2 + r_2 r_3 + \cdots + r_n r_1} \mod p,$$

where, again, the indices are taken in a cycle.

In addition to requiring the generation of t random integers in $\mathbb{Z}_q$, this protocol requires computing a total of t inverses, t^2 exponentiations, and $t^2 + t$ multiplies in $\mathbb{Z}_p$.

B. Complexity Comparison

Regardless of the master key loading scheme, our protocol requires at most $2(t-1)$ PRF evaluations. Assuming that an X-bit secure HMAC is used as the PRF in our protocol and that a hash-based pseudo-random number generator (e.g., [43]) is used in Burmester and Desmedt's protocol, the total complexity of the PRF evaluation and random number generation in the two protocols is comparable. The difference in computational complexity therefore lies elsewhere.

By extending the standard linear time greedy set cover algorithm [40] to the hypergraph setting, it can be shown that the number of operations required by the MCSH approximation in Algorithm 1 grows as $O(tn)$ (assuming $O(n)$ total master keys). Since this algorithm is executed at all t clients in parallel, the total complexity is $O(t^2 n)$. Owing to the complexity of modular arithmetic over very large integers, this complexity growth compares favorably with the BD protocol. Specifically, the complexity of inversion, multiplication, and exponentiation in $\mathbb{Z}_p$ grows with the modulus p as $O\left((\log p)^2\right)$, $O\left((\log p)^2\right)$, and $O\left((\log p)^3\right)$, respectively. Thus, the complexity of the X-bit secure BD protocol grows as $O(t^2 X^3)$.

VII. Conclusion and Future Work

Motivated by a desire to employ public key cryptography for group communications in ad hoc and sensor networks, this paper described a protocol that can establish a group key among t clients using far fewer than t transmissions. When master keys are distributed randomly in the network, our protocol requires $O(\log_b t)$ public multicast transmissions, where $1 - 1/b$ is the probability that a given client possesses a given master key. The group key established by our protocol is secure in the information-theoretic sense against out-of-network eavesdroppers; however, it may be exposed to *undetected* malicious in-network clients. This vulnerability, which is common to all protocols that use master keys, may be a reasonable price to pay for increasing the energy-efficiency of group key agreement in many operational scenarios.

Our approach was inspired by recent work by the first two authors. In [15], the amount of communication required to generate a key of prescribed length was characterized in the combinatorial setting (i.e., when nodes share master keys according to some arbitrary distribution). When a group key with the same length as the master keys is desired, obtaining an optimal communication scheme is equivalent to solving the Minimum Connected Subhypergraph problem on the hypergraph implied by the master key distribution. This problem is NP-hard. We therefore employ a greedy approximation in our protocol that provides a strong performance guarantee.

Future work will address the translation of the abstract protocols described in this paper to an Application Layer solution suitable for implementation in power-constrained wireless networks. The key issue to address in that translation is protocol scalability. For example, a pseudo-random master key distribution that can be derived from a single seed could be used in place of the random distributions considered herein. This would enable low-overhead network join operations while maintaining energy-efficient group key agreement. Other issues to address include support for group join operations, robustness to lossy wireless links, and key refreshing.

Appendix

In [44], Ren and Zhao studied a generalization of the Minimum Connected Subhypergraph problem which we review briefly in this appendix in order to prove Result 2.

Let V be a finite set, let $E = \{e_i \subseteq V\}_{i \in I}$ be a collection of subsets of V, and let G be a connected graph with vertex set E. A *connected set cover* $F \subseteq E$ with respect to (V, E, G) is a set cover of V such that F induces a connected subgraph of G. If G is a complete graph, then the Minimum Connected Set Cover (MCSC) problem is equivalent to Set Cover.

In order to state Ren and Zhao's greedy algorithm for approximating MCSC, some notation is required. For $e_j, e_k \in E$, $d_G(e_j, e_k)$ is the length of the shortest path between e_j and e_k in G. The sets e_j and e_k are *graph-adjacent* if $d_G(e_j, e_k) = 1$ and *cover-adjacent* if $e_j \cap e_k \ne \emptyset$. The *cover-diameter* $D_c(G)$ is the maximum distance in G between any two cover-adjacent sets. For any $\emptyset \ne F \subset E$ and $g \in E \setminus F$, an *$(F \to g)$-path* is a path $\{e_{i_0}, e_{i_1}, \ldots, e_{i_k}\}$ in the graph G such that $e_{i_0} \in F$, $e_{i_k} = g$, and $e_{i_1}, \ldots, e_{i_k} \in E \setminus F$. Finally, the *weight ratio* of an $(F \to g)$-path is defined as the length of the path in G divided by the number of elements of V that are covered by $e_{i_1} \cup \cdots \cup e_{i_k}$ but not by the union of the sets in F.

In [44, Theorem 1], Ren and Zhao proved that Algorithm 3 yields a connected set cover F with a size that is at most

$$D_c(G)\,(1 + H(\gamma - 1)) \tag{11}$$

times as large as the optimal MCSC solution, where γ is the size of the largest element in S and $H(\gamma) = \sum_{k=1}^{\gamma} 1/k$ is the γ^{th} harmonic number. Moreover, they proved that this approximation ratio is order-optimal unless NP contains slightly superpolynomial algorithms.

It is readily shown that the MCSH problem is a special case of the MCSC problem where G is defined such that the vertices corresponding to sets e_j and e_k are connected via an edge if and only if $e_j \cap e_k \neq \emptyset$. In this case, cover-adjacency and graph-adjacency are equivalent and every iteration of the while loop in Algorithm 3 selects a path $\{e_{i_0}, e_{i_1}\}$ that maximizes $|e_{i_1} \setminus U|$. This is the same heuristic used to choose the next occupancy set in Algorithm 1. Moreover, the heuristic used to choose the first element g in Algorithm 3 is identical to that used to choose the first occupancy set in Algorithm 1. Therefore, the greedy approximation for the MCSH problem used in Algorithm 1 is a special case of Ren and Zhao's algorithm. The proof of Result 2 is completed by noting that $D_c(G) = 1$ when cover-adjacency and graph-adjacency are equivalent.

Algorithm 3 Ren and Zhao's greedy algorithm for the Minimum Connected Set Cover problem.

Input: (V, E, G)
Output: A connected set cover F.

Choose $g \in E$ such that $|g|$ is the maximum, and let $F = \{g\}$ and $U = g$.
while $V \setminus U \neq \emptyset$ **do**
  $\mathcal{P}_F \leftarrow \emptyset$;
  for $g \in E \setminus F$ **do**
    if *g is cover-adjacent or graph-adjacent to any set in F* **then**
      $\mathcal{P}_F \leftarrow \mathcal{P}_F \cup \{P_{F,g}\}$, where $P_{F,g}$ is a *shortest* $(F \to g)$-path;
    end
  end
  $\{e_{i_0}, e_{i_1}, \ldots, e_{i_k}\} \leftarrow$ path in $\mathcal{P}_F$ with minimum weight ratio;
  $F \leftarrow F \cup \{e_{i_0}, e_{i_1}, \ldots, e_{i_k}\}$,
  $U \leftarrow U \cup e_{i_0} \cup e_{i_1} \cup \cdots \cup e_{i_k}$;
end

REFERENCES

[1] T. R. Halford, T. A. Courtade, and K. M. Chugg, "Energy-efficient, secure group key agreement for ad hoc networks," in *Proc. IEEE Commun. Netw. Security Conf.*, Washington, DC, USA, Oct. 2013, pp. 181–188.

[2] P. Sakarindr and N. Ansari, "Security services in group communications over wireless infrastructure, mobile ad hoc, and wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 8–20, Oct. 2007.

[3] K. Ren, W. Lou, B. Zhu, and S. Jajodia, "Secure and efficient multicast in wireless sensor networks allowing ad hoc group formation," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2018–2029, May 2009.

[4] T. R. Halford, "Secure many-to-some communications," in *Proc. IEEE Mil. Commyn. Conf.*, San Diego, CA, USA, Nov. 2013, pp. 243–247.

[5] A. C. Atici, L. Batina, J. Fan, I. Verbauwhede, and S. B. Ors, "Low-cost implementation of NTRU for pervasive security," in *Proc. Int. Conf. Appl.-Spec. Syst., Architect. Process.*, Leuven, Belgium, Jul. 2008, pp. 79–84.

[6] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE Conf. Comput. Commun.*, Tel Aviv, Israel, Mar. 2000, pp. 585–594.

[7] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 8, pp. 769–780, Aug. 2000.

[8] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, and G. Tsudik, "Secure group communication using robust contributory key agreement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 5, pp. 468–480, May 2004.

[9] S. Jarecki, J. Kim, and G. Tsudik, "Flexible robust group key agreement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 879–886, May 2011.

[10] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Berlin, Germany: Springer-Verlag, 2003.

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

[12] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Advances in Cryptology—EUROCRYPT*, vol. 950. Berlin, Germany: Springer-Verlag, 1995, pp. 275–286, ser. Lecture Notes in Computer Science.

[13] J.-W. Ho, M. Wright, and S. K. Das, "ZoneTrust: Fast zone-based node compromise detection and revocation in wireless sensor networks using sequential hypothesis testing," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 4, pp. 494–511, Jul./Aug. 2012.

[14] T. A. Courtade and R. D. Wesel, "Coded cooperative data exchange in multihop networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1136–1158, Feb. 2014.

[15] T. A. Courtade and T. R. Halford, "Coded cooperative data exchange for a secrect key," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014, pp. 776–780.

[16] T. T. Mapoka, "Group key management protocols for secure mobile multicast communication: A comprehensive survey," *Int. J. Comput. Appl.*, vol. 84, no. 12, pp. 28–38, Dec. 2013.

[17] B. E. Jung, "An efficient group key agreement protocol," *IEEE Commun. Lett.*, vol. 10, no. 2, pp. 106–107, Feb. 2006.

[18] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3047–3061, Dec. 2004.

[19] C. Chan, "Linear perfect secret key agreement," in *Proc. IEEE Inf. Theory Workshop*, Paraty, Brazil, Oct. 2011, pp. 723–726.

[20] M. Mukherjee and N. Kashyap, "On the communication complexity of secret key generation in the multiterminal source model," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014, pp. 1151–1155.

[21] S. Nitinawarat and P. Narayan, "Perfect omniscience, perfect secrecy and Steiner tree packing," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6490–6500, Dec. 2010.

[22] H. Tyagi, "Common information and secret key capacity," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5627–5640, Sep. 2013.

[23] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, and A. Reznik, "Secret key generation for a pairwise independent network model," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6482–6489, Dec. 2010.

[24] H. Tyagi, N. Kashyap, Y. Sankarasubramaniam, and K. Viswanathan, "Fault-tolerant secret key generation," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 1787–1791.

[25] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, Oct. 1986.

[26] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. IEEE Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[27] M. Gonen and M. Langberg, "Coded cooperative data exchange problem for general topologies," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2606–2610.

[28] T. A. Courtade and R. D. Wesel, "Weighted universal recovery, practical secrecy, and an efficient algorithm for solving both," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 2011, pp. 1349–1357.

[29] A. Scaglione and Y.-W. Hong, "Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2082–2092, Aug. 2003.

[30] T. R. Halford and K. M. Chugg, "Barrage relay networks," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2010, pp. 1–8.

[31] F. Ferrari, M. Zimmerling, L. Mottola, and L. Theile, "Low-power wireless bus," in *Proc. ACM Conf. Embedded Netw. SenSys*, Toronto, ON, Canada, Nov. 2012, pp. 1–14.

[32] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, "Deterministic algorithm for the cooperative data exchange problem," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011, pp. 410–414.

[33] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proc. ACM Conf. Comput. Commun. Security*, Athens, Greece, Nov. 2000, pp. 235–244.

[34] L. Chen, *NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*. Gaithersburg, MD, USA: National Institute of Standards and Technology, Oct. 2009.

[35] S. Ruj, A. Nayak, and I. Stojmenovic, "Pairwise and triple key distribution in wireless sensor networks with applications," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2224–2237, Nov. 2013.

[36] C. Vercellis, "A probabilistic analysis of the set covering problem," *Ann. Oper. Res.*, vol. 1, no. 3, pp. 255–271, 1984.

[37] O. A. Telelis and V. Zissimopoulos, "Absolute $o(logm)$ error in approximating random set cover: An average case analysis," *Inf. Process. Lett.*, vol. 94, no. 4, pp. 171–177, May 2005.

[38] A. Demers *et al.*, "Epidemic algorithms for replicated database maintenance," in *Proc. ACM Symp. Principles Distrib. Comput.*, Vancouver, BC, Canada, Aug. 1987, pp. 1–12.

[39] A. S. Akyurek and E. Uysal-Biyikoglu, "A low complexity distributed algorithm for computing minimum-depth multicast trees in wireless networks," in *Proc. IEEE Mil. Commun. Conf.*, San Jose, CA, USA, Nov. 2010, pp. 1918–1923.

[40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2002.

[41] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications*. Boston, MA, USA: Birkhaüser, 1998, pp. 547–566.

[42] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[43] E. Barker and J. Kelsey, *NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. Gaithersburg, MD, USA: National Institute of Standards and Technology, Jan. 2012.

[44] W. Ren and Q. Zhao, "A note on 'Algorithms for connected set cover problem and fault-tolerant connected set cover problem'," *Theor. Comput. Sci.*, vol. 412, no. 45, pp. 6451–6454, Oct. 2011.

Thomas R. Halford received the B.A.Sc. degree in engineering physics from Simon Fraser University, Burnaby, BC, Canada, in 2001, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2007. From 2007 to 2015, he was with TrellisWare Technologies, Inc., San Diego, CA. He currently leads technology and business development at WPL, Inc. in Manhattan Beach, CA. His research interests span all aspects of military communications. He served as a track co-chair at the 2015 Military Communications Conference.

Thomas A. Courtade (S'06–M'13) received the B.Sc. degree (summa cum laude) in electrical engineering from Michigan Technological University, Houston, MI, USA, in 2007 and the M.S. and Ph.D. degrees from UCLA in 2008 and 2012, respectively. He is an Assistant Professor in the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley, CA, USA. Prior to joining UC Berkeley in 2014, he was a Postdoctoral Fellow supported by the NSF Center for Science of Information. His honors include a Distinguished Ph.D. Dissertation award and an Excellence in Teaching award from the UCLA Department of Electrical Engineering, and a Jack Keil Wolf Student Paper Award for the 2012 International Symposium on Information Theory.

Keith M. Chugg (S'88–M'95–F'10) received the B.S. degree (high distinction) in engineering from Harvey Mudd College, Claremont, CA, USA, in 1989 and the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, CA, in 1995. He is currently a Professor in the Ming Hsieh Department of Electrical Engineering at USC.

His research interests include communications, signal processing, graphical models, and digital hardware implementation. He is co-author of the book *Iterative Detection: Adaptivity, Complexity Reduction, and Applications* (Kluwer Academic Press). He is a co-founder of TrellisWare Technologies, Inc., where he is Chief Scientist. He was awarded the ASEE Frederick Emmons Terman Award in 2008 and the Fred Ellersick Award as the outstanding unclassified paper at MILCOM 2003.

Xiaochen Li received the B.S. and M.E. degrees in electronic engineering from Peking University, Beijing, China, in 2002 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2009. From 2009 to 2010, she was a Research Associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. Since 2011, she has been with TrellisWare Technologies, Inc. in San Diego, CA, USA. Her research interests include wireless military communication, resource allocation for delay sensitive networks, and cross layer optimization.

Gautam Thatte received the B.S. degree in Engineering from Harvey Mudd College, Claremont, CA, USA, in 2003, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2010. He is currently a Systems Engineer at TrellisWare Technologies, Inc. His research interests include estimation and detection in decentralized networks, multi-user detection, and applications of iterative detection and interference mitigation in military communications systems.