A Fast and Efficient Source Authentication Solution for Broadcasting in Wireless Sensor Networks

Taojun Wu, Yi Cui, Brano Kusy, Akos Ledeczi, Janos Sallai, Nathan Skirvin, Jan Werner, and Yuan Xue*

Institute for Software Integrated Systems (ISIS) and EECS, Vanderbilt University
Email: {taojun.wu, yuan.xue, nathanael.skirvin, jan.werner, branislav.kusy, janos.sallai, akos.ledeczi, yi.cui}@vanderbilt.edu

Summary. Wireless sensor network has drawn increasing attentions in recent years due to its wide range of applications. Often deployed in hostile environments, wireless sensor network is particularly vulnerable to malicious attacks. Thus security becomes a critical issue. This paper studies the security support for source authentication for broadcasting in wireless sensor networks. Our problem is motivated by a real sensor network application scenario – Dirty Bomb Detection and Localization, which requires efficient broadcast source authentication service in real-time. Although there exist broadcast source authentication solutions developed for wireless sensor networks, they either require significate latency in key release from a one-way hash key chain, or need a large memory space and/or involve high communication overhead. None of these solutions could meet the strict requirements from real-time communication and the limited memory space in our application.

To address this issue, we present a broadcast source authentication mechanism based on multiple message authentication codes (MultiMAC). The novel contribution of this work is that it proposes a deterministic combinatorial key distribution scheme that provides scalable authentication service with limited key storage need. This authentication service is implemented as a security component in TinyOS as part of the Dirty Bomb Detection and Localization application, where its performance is validated.

1 Introduction

The convergence of micro-eletro-mechanical system technology, wireless communication and digital electronics leads to the emergence of wireless sensor

* This work was supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: Cisco, ESCHER, HP, IBM, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia and United Technologies.

networks [1], which are capable of sensing, data processing, and communicating. Sensor networks can be readily deployed in diverse environments to collect and process useful information in an autonomous manner. Thus, they have a wide range of applications in the areas of health care, military, and disaster detection.

Often deployed in hostile environments, wireless sensor networks are vulnerable to a variety of attacks [2, 3]. Thus security becomes a critical issue to ensure safe operation of wireless sensor network. Existing research has provided a variety of security supports (e.g., protection of data confidentiality and integrity) for wireless sensor networks. Representative works include link layer security architecture [4], secure routing [2, 3], and key management and distribution mechanisms [5, 6].

In this paper, we study the problem of source authentication for broadcast[2] traffic in wireless sensor networks. Our problem is motivated by the need from a real sensor network application scenario – Dirty Bomb Detection and Localization [7]. In this application, the master sensor, carried by a moving policeman, will broadcast the localization command to the rest of the sensors via multi-hop wireless communication so that they could start the synchronization and localization operation simultaneously. In this application, every round of localization operation, including communication delay, needs to finish in less than 3 seconds. This application also requires each receiver sensor node to be able to authenticate the broadcast message (i.e., localization command) from the source (the master sensor). Acceptance of false command will trigger unnecessary synchronization and localization operations, which waste scarce battery energy and cause operation confusion among sensors in the worst case.

There exist two catagories of solutions for broadcast source authentication: asymmetric-key-based and symmetric-key-based mechanisms. In wired network, asymmetric-key-based solution is a popular approach due to its convenience. However, its high computation overhead becomes a huge obstacle for its application in wireless sensor networks. The work of [8, 9] have studied the fitness of the most popular asymmetric key algorithms including RSA and ECC in wireless sensor networks. Through advanced implementation technologies, their study shows that the cryptographic operations of ECC algorithm are viable when the computational power and space of the sensor node are dedicated to security operations. However, such an asymmetric-key-based solution could not be applied to our application scenario, as most storage space and computation power of the sensors are used for the detection and localization functions, leaving little space for security functions.

In $\mu TESLA$ [10], the authors present a symmetric-key-based broadcast source authentication scheme. This scheme is further extended in BABRA [11]. The basic idea of these two schemes is to create an *asymmetry* in time among

[2] In this paper, broadcast refers to the flooding messages that may be forwarded through multiple hops, instead of the local broadcast message.

the broadcasting source and the receivers through the delayed disclosure of key. In particular, the source node will generate a one-way key chain by applying a hash function iteratively from the last key. The keys will be applied to generate message authentication codes (MAC) sequentially, but will be released with certain delay after the packets are received. As a consequence, the receivers are unable to authenticate the messages they receive immediately. This approach introduces time latency in authentication, thus is not applicable in real-time systems.

Pair-wise keys and their establishment have received extensive research [12, 13, 14, 15, 16, 6, 17]. Once established, pair-wise key could also be used to provide broadcast authentication service either through generated session key or unique key-path. Yet these two approaches will either involve high delay and computational overhead in repeatedly generating and verifying MAC, or high communication overhead in generation of session keys that prevents it to scale to large networks.

To address the above issues and meet the real-time and efficient authentication need from our application, we propose a novel broadcast source authentication mechanism based on multiple message authentication codes (MultiMAC). This mechanism requires the sensor nodes to have different yet overlapping set of keys (so-called key ring). To authenticate a message, the source node will generate a list of MACs based on its keys, and append them to the message. The receiver node will verify the message based on the MACs which are generated using the keys that are shared with the source. To support such an authentication service over the sensors with limited storage space, the key ring has to be designed to satisfy a set of conditions. We formulate this problem as a combinatorial problem and present a deterministic combinatorial key distribution scheme that supports scalable authentication service for wireless sensor networks.

The major contributions of this work are summarized as follows. Theoretically, it integrates a hierarchical key structure with the deterministic combinatorial key distribution schemes to support large-scale sensor network broadcast authentication services. Practically, it implements the proposed authentication solution as a security module in TinyOS as part of the Dirty Bomb Detection and Localization system and validates its performance through the integrated system experiment. It is also worth noting that although our MultiMAC scheme is designed for the Dirty Bomb Detection and Localization system, it could be applied to general broadcast scenarios in wireless sensor networks.

The rest of the paper is organized as follows: Section 2 explains our symmetric-key-based broadcast source authentication approach. A key predistribution scheme is next presented in Section 3 that scales to large sensor networks in a deterministic way. The detailed protocol implementation of our approach as well as field measurement results are included in Section 4. Section 5 concludes the paper and points out some future work.

2 Symmetric-Key-Based Broadcast Source Authentication Model

We investigate the source authentication problem for broadcast in wireless sensor networks based on symmetric key mechanisms. Here are two benchmark scenarios of broadcast: one scenario involves only a single sender; the other scenario allows every node to be the source of broadcast. Our research focuses on the second scenario while the first scenario could be straightforwardly addressed based on our approach.

Inspired by the work of [18, 19], the basic idea of broadcast source authentication in wireless sensor networks works as follows. Each sensor node in the network $i \in N$ has a *different* set of keys S_i called *key ring*. To authenticate message M, the sender node i will generate a *message authentication code* $MAC(K_i^j, M)$ using each key in its key ring $K_i^j \in S_i$. The full collection of MACs $MAC(K_i^1, M)||MAC(K_i^2, M)||\ldots||MAC(K_i^l, M)$ will be transmitted together with the message M. Each recipient node $r \in N$ verifies all the MACs that are created using the common keys which it shares with the sender, *i.e.*, the keys in set $S_i \cap S_r$ where S_r is the key ring of r. If any of these MACs is incorrect, then r rejects the message. Fig. 1 illustrates a simple example of this idea, where sensors A, B, C, D have different combinations of overlapping keys, and sensor A tries to imposter C. When sensor A sends out a message M, it attaches $MAC(1, M)||MAC(4, M)$ to it. The other recipients of the message then verify MACs generated with any common keys they are sharing with the sender. However, this key pre-distribution does not assure sufficient authentication and we would discuss later the conditions necessary to achieve authentication with shared keys.

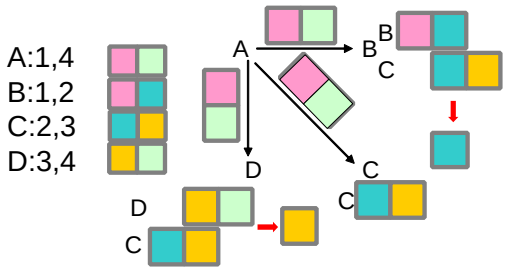

Fig. 1. Simple example of shared key authentication.

In comparison with the existing solutions for wireless sensor network broadcast authentication such as $\mu TESLA$ [10] and [11], our approach offers latency-free authentication – the recipient nodes are able to verify the message immediately without waiting for key disclosure. Further without using delayed key disclosure, this mechanism does not require any clock synchro-

nization among sensor nodes, which itself is a hard problem [20] and can be susceptible to attacks.

The major challenge to apply the presented mechanism to broadcast authentication is the key pre-distribution. There are two major issues involved: (1) whether the key assignment scheme supports authentication of any sensor node as source of broadcast; (2) whether the scheme is scalable to larger systems. These issues become even more challenging when the application system enforces a strict extra delay bound. Our solution to this problem is one of the major contribution of this paper. Next in this section we will present some conditions on key distribution.

Table 1. Notations used in Sec. 2

m	number of distinct keys
n	number of wireless sensors
N_i	the i-th sensor
S_i	the i-th subset of keys, assigned to N_i
Key_j	the j-th key
$Group_j$	the j-th group of sensors, sharing the j-th secret key
k_i	$\|S_i\|$, number of keys in S_i
d_{ij}	$\|S_i \cap S_j\|$, number of common keys between S_i and S_j

Generally speaking, the key grouping scheme for any network needs to fulfill the following **Baseline Grouping** conditions.

1. *Non-empty subset condition*: any sensor N_i has at least one key, i.e. $\forall S_i, |S_i| = k_i \geq 1$.
2. *Unique subset condition*: no two sensors have the same subset of keys, i.e., $\forall i, \forall j, j \neq i$ implies $S_i \neq S_j$. Otherwise, the i-th sensor can easily imposter the j-th node.

Furthermore, the grouping of sensor nodes in a network needs to satisfy **Authentication Feasibility** conditions below in order for the grouping scheme to provide multi-hop broadcast source authentication.

1. *Subset overlapping condition*: any subset of keys S_i assigned to sensor N_i overlaps with at least one other subset of keys S_j of some sensor N_j, i.e., $\forall i, \exists j \neq i,\ s.t.\ |S_i \cap S_j| = d_{ij} \geq 1 (S_i \cap S_j \neq \Phi)$. The *subset overlapping condition* ensures any sensor can be authenticated by at least one other sensor. In other words, for any possible identity spoofing, there will exist some sensor in the system to detect the activity.
2. *Overlapping distinction condition*: the sets of overlapping keys of any subset of keys S_i with any two other key subsets S_j, S_k are distinguishable, i.e., $\forall i, \forall j, \forall k \neq j, S_i \cap S_j \nsubseteq S_i \cap S_k$.

In order for this group-based authentication scheme to work, the grouping parameters of sensors mentioned above need to satisfy the following constraints:

1. *Key number bound condition*: the number of keys in any subset is bounded by total number of keys in key pool. $d_{ij} \leq k_i, k_j \leq m$;
2. *Sensor number condition*: the number of obtained different key subsets needs to exceed the number of sensors. In many occasions, it is convenient to have equal sized subsets, i.e. $\forall i, k_i = k^*$. In this case, a bottom-line version of this constraint becomes $\binom{m}{k^*} \geq n$, which ensures the number of possible combinations of k^* keys larger than sensor number n.

Additionally, some similar works have assumed stronger constraints to achieve their goals of establishing single secret key for any pair of sensors. Some typical constraints are:

1. *Equal-size overlapping condition*: any sensor N_k has equal sized overlapping set of keys with all other sensors. $\forall i, \forall j \neq i, d_{ij} = d^*$;
2. *Unique overlapping condition*: any sensor N_k has unique overlapping set of keys with all other sensors. This can be expressed as: $\forall i, j, \nexists k,\ s.t.\ S_i \cap S_k = S_k \cap S_j$. An even stronger version of this condition requires no two overlapping combination of keys are the same, in other words: $\forall i, j, k, l,\ s.t.\ S_i \cap S_j \neq S_k \cap S_l$;

3 Key Pre-distribution Scheme

In this section we present the key pre-distribution scheme for our symmetric-key-based broadcast source authentication. Our solution starts with a base scheme and it is afterwards extended in a hierarchical way to scale up to larger sensor networks.

3.1 Base Key Distribution Scheme

Our key pre-distribution scheme takes the advantage of existing research in combinatorics. Generally speaking, a working key distribution scheme needs to satisfy **Authentication Feasibility** conditions listed in Sec. 2. These define a combinatorial problem. According to Colbourn et al. [21], schemes satisfying these conditions are known for most $n \leq 50$.

The complexity in finding satisfying schemes grows rapidly [21] when n increases. To support large-scale sensor networks, we choose following grouping scheme ($n = 7$) as our *BaseScheme* and extend it with a hierarchical structure.
$BaseScheme = \{(1,2,4,7),(1,2,5,6),(1,3,4,6),(1,3,5,7),(2,3,4,5),(2,3,6,7),(4,5,6,7)\}$.

3.2 Extension of Base Scheme

For ease of presentation, before explaining how to extend the base scheme, we denote $KeyScheme = \{a, b, c, d, e, f, g\}$, where $a, b, \ldots, g$ are called $KeyBundle$s which are key combinations following $BaseScheme$. For example, if we have a set of keys $KeySet = \{Key_1, Key_2, \ldots, Key_7\}$, we can map element a to the first $KeyBundle_1 = (Key_1, Key_2, Key_4, Key_7)$. Therefore, if we have two distinct key sets (KS_1, KS_2), and $KS_1 \cap KS_2 = \Phi$, we can have two different key schemes $KeyScheme_1 = \{a, b, \ldots, g\}$ and $KeyScheme_2 = \{A, B, \ldots, G\}$.

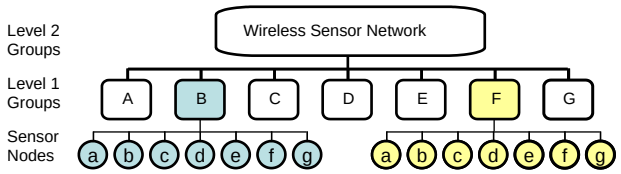

Fig. 2. Illustration of Extending Base Scheme to Support $n = 49$.

First of all, the n sensors of the network is divided into $GroupNum$ groups of size 7. Within each of these groups, the nodes are assigned one of the seven key bundles (a combination of keys chosen from seven keys in the key pool) out of $BaseScheme$. In this way, any nodes within the same group can authenticate mutually. However, to enable broadcast authentication among nodes from different groups, more efforts are needed. In the next stage, seven new keys are introduced. All the nodes in the same group are treated as one single parent node, and the series of groups generated in the first step is regarded as a higher level network consisted of parent nodes. These parent nodes are again separated into parent groups, each containing 7 parent nodes, or 49 sensor nodes. In a similar way, key bundles organized according to $BaseScheme$, consisting of new keys introduced during this stage, are assigned to parent nodes within these parent groups. Consequently, all sensor nodes of the same parent group now have four additional keys, while maintaining their four old keys obtained during stage one. The previous two steps would recursively repeat until $GroupNum \leq 7$.

The Exclusion Basic System (EBS) presented in [22] also provides a key management scheme for multicast in wireless sensor networks. The major difference of our combinatorial deterministic approach and EBS is that our scheme satisfies the **Authentication Feasibility** condition. This eliminates the need of establishing session keys before actual authentication for EBS case. The relatively small and predictable size of key ring also distinguishes our scheme. For example, when $n = 49$, our scheme only needs 8 keys per sensor $(4log_7 n)$.

4 Protocol Implementation and Performance Evaluation

We implement and evaluate our protocol as a nesC component (MultiMAC) under TinyOS 1.1.15 in the real system "Dirty Bomb Detection and Localization" [7]. The system tracks a radiation detector as it traverses an area where a network of sensor nodes (Mica2 motes) has been deployed.

The key grouping scheme we use is illustrated in Table 2, following discussions of Sec. 2. For compatible reason and to save storage space, we use the SkipJack implementation provided in TinySec [4] as our symmetric cipher to compute and verify MACs. As shown in Table 2, every sensor node in the system stores a different key ring in its ROM. Multiple MACs of every outgoing message are calculated, using the key ring assigned to the sending mote. The receiving mote authenticates the message by recomputing MACs using its shared keys with the sender. Our protocol fits into contingent real-time constraints of the system and is discussed in detail next.

Table 2. Key Distribution Scheme Used for Implementation

Node #	Assigned Key Ring
0	[1, 2, 4, 7, 8]
1	[1, 2, 5, 6, 8]
2	[1, 3, 4, 6, 8]
3	[1, 3, 5, 7, 8]
4	[2, 3, 4, 5, 8]
5	[2, 3, 6, 7, 8]
6	[4, 5, 6, 7, 8]
7	[1, 2, 4, 7, 9]
8	[1, 2, 5, 6, 9]
9	[1, 3, 4, 6, 9]
10	[1, 3, 5, 7, 9]
11	[2, 3, 4, 5, 9]
12	[2, 3, 6, 7, 9]
13	[4, 5, 6, 7, 9]

4.1 Protocol Procedure

Our broadcast authentication scheme operates in following steps:

1. The key ring for every sensor N_i is initialized to one out of n subset of keys, S_i;
2. A key mapping function (or structure) exists in every sensor N_i so that the detailed key grouping is available locally and N_i can get to know subset of keys S_j for any sensor N_j;

3. When a sensor N_i sends out messages, it appends multiple MACs to the message. Each MAC is computed with a key in S_i, hence there are k^* MACs in total;
4. When a sensor N_j receives a message from N_i, it checks to find its common keys with N_i and then verifies if the corresponding MACs are correct. N_j will "reject" the message if the provided multiple MACs contain any wrong MAC. Otherwise, it will "accept" it;

4.2 Reducing Length of Multiple MACs

One obvious problem with our protocol design is that our scheme requires k^* times as many bytes for single message authentication. The *Bloom filter* structure can serve as a possible solution to this problem. In [23], Ye et al. proposed to use several hash functions to transform a bunch of MACs from various sensors to a combinated Bloom filter of equal size. This bit string then serves as one MAC. Similarly, we transform individual MAC to a smaller space and obtain a shorter MAC. In this way, each key will still have a designated portion in the final MACs, although weakened.

4.3 Broadcast Authentication Overhead

We will now present the measurement results from the system. Our MultiMAC component requires a reasonable amount of memory storage. Table 3 shows that the storage requirements scale gracefully with larger key rings, as larger key rings simply use more memory to store additional keys. The memory requirements imposed by MultiMAC are acceptable for the "Dirty Bomb Detection and Localization" system.

Table 3. Memory Usage

# of Keys	ROM (bytes)	RAM (bytes)
2	2778	105
3	2814	153
4	2824	201
8	2894	393
16	3022	777
32	3278	1545

We next discuss overhead of computating single MAC of different length. Table 4 shows the time (in milliseconds) required to compute a single MAC for various MAC lengths (in bytes). As the result shows, generating MACs of different lengths does not affect processing time much.

As a receiving sensor can share varying number of keys with the sending mote, we finally measure the time (in milliseconds) of verifying different number of MACs for a single message. This is shown in Table 5. In this table, the

Table 4. Single MAC Processing Time

MAC Length (bytes)	Compute Time (ms)
2	3.42
3	3.45
4	3.51

Table 5. Multiple MACs Processing Time

Number of MACs	Verify Time (ms)
0	<0.1
1	1.3
2	2.5
3	3.7

number of MACs represents the number of shared keys between the broadcasting source node and the receiving node. The verification time is almost linear to the number of MACs needing verification, as expected.

5 Conclusions and Future work

In this paper we have presented a fast and efficient broadcast source authentication protocol in a real-time system: "Dirty Bomb Detection and Localization". We address scalability issue of existing deterministic approaches by applying hierarchical structure to combinatorial results. The resulting key predistribution scheme ensures efficient authentication and the measured results confirm us of the efficiency and low overhead of our approach. Our nesC implementation of MultiMAC for wireless sensor networks under TinyOS fulfils its security demands and satisfies the contingent time constraints.

For future work we are interested in adding re-keying and key revocation mechanisms. These are essential for the system to operate properly if we want to add new sensor nodes and remove faulty ones after deploying it. Solution to this problem might require adding extra information to current key mapping function mentioned in second step of protocol procedure (Sec. 4).

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cyirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
2. C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, no. 2–3, pp. 293–315, September 2003.

3. Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, 2003, pp. 1976–1986 vol.3. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1209219
4. C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 162–175.
5. W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 42–51.
6. H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 197.
7. "Dirty bomb detection and localization," http://www.isis.vanderbilt.edu/Projects/rips/.
8. D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *SECON 2004. First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, October 2004.
9. N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *CHES's 2004 Workshop on Cryptographic Hardware and Embedded Systems*, Aug 2004, pp. 119–132.
10. A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.
11. Y. Zhou and Y. Fang, "Babra: Batch-based broadcast authentication in wireless sensor networks," in *Proc. of IEEE GLOBECOM*, Nov 2006.
12. R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 335–338.
13. L. Zhou, J. Ni, and C. V. Ravishankar, "Efficient key establishment for group-based wireless sensor deployments," in *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, 2005.
14. J. Lee and D. R. Stinson, "Deterministic key predistribution schemes for distributed sensor networks," in *Selected Areas in Cryptography*, 2004, pp. 294–307.
15. D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2003, pp. 72–82.
16. L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 41–47.
17. W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, 2004.

18. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in *Proc. IEEE INFOCOM'99*, vol. 2. New York, NY: IEEE, Mar. 1999, pp. 708–716.
19. P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication," in *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1999, pp. 93–100.
20. M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2005, pp. 107–116.
21. C. J. Colbourn and J. H. Dinitz, "Graphical designs," in *The CRC Handbook of Combinatorial Designs*. Boca Raton: CRC Press, 1996, pp. 367–369.
22. L. Morales, I. H. Sudborough, M. Eltoweissy, and M. H. Heydari, "Combinatorial optimization of multicast key management," in *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9*. Washington, DC, USA: IEEE Computer Society, 2003, p. 332.2.
23. F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004, pp. 2446–2457.