

Authenticated Queries in Sensor Networks

Zinaida Benenson

Department of Computer Science,
RWTH Aachen University, Germany

Abstract. This work-in-progress report investigates the problem of *authenticated querying* in sensor networks. Roughly, this means that whenever the sensor nodes process a query, they should be able to verify that the query was originated by a legitimate entity. I precisely define authenticated querying, analyze the design space for realizing it and propose solutions to this problem in presence of *node capture* attacks.

1 Introduction

Consider a sensor network which is deployed over a large geographic area. The maintainer of the sensor network offers services to the users: They can post queries to the sensor network using some mobile device. In this case, only the queries of legitimate users should be answered by the network. However, existing query processing systems (for an overview, see e.g. [26]) are not concerned with this issue. Meanwhile, this problem becomes especially difficult in presence of node capture attacks.

1.1 Node Capture in Sensor Networks

Node capture means gaining full control over a sensor through a physical attack, e.g., opening the sensor's cover and reading out its memory and changing its program. A node capture attack can only be mounted on a small portion of the network if the network is sufficiently large, as direct physical access is needed.

This type of attack is fundamentally different from gaining control over a sensor remotely through some software bug, e.g., a buffer overflow. As all sensors are usually assumed to run the same software, in particular, the same operating system, finding an appropriate bug would allow the adversary to control the whole sensor network.

1.2 Motivation for Authenticated Querying: An Example

Directed Diffusion [12], a popular paradigm for organizing sensor networks, allows the user to post queries at any arbitrary sensor node (called the *sink*). The sink then floods the network with the query. After some time, sensor nodes start sending their aggregated data towards the sink. The sink gives the data to the user. In this case, to prevent the adversary from querying the sensor network, an access control mechanism should be built into each sensor node.

Consider an adversary who wants to gain unauthorized access to the data. He can try either to subvert the access control mechanism, or to find some weaker point in the sensor network architecture. For example, if the communication between the sensors happens without encryption and authentication, the adversary would bypass access control mechanism by directly attacking the communication protocol (eavesdrop, insert his own messages).

But even if all communication between the sensors is properly encrypted and authenticated, access control remains a separate problem which has to be solved. To illustrate this, consider a sensor network with Directed Diffusion mechanism where the sink is able to organize secure and authenticated communication with other sensor nodes.¹

Suppose that, additionally to secure authenticated communication with the sink, some access control mechanism is built into each sensor node. This even could be an SSL/TLS-like protocol, as Gupta et al. [10] recently showed. Their implementation of SSL handshake on extremely resource constrained MICA2 sensors [5] takes less than 4 seconds.

However, if the adversary can disable the access control mechanism on a single sensor node, for example by capturing it, he would be able to query the entire sensor network. This single sensor, acting as a *new* sink, will build a secure authenticated channel to other sensor nodes, but this would not prevent the adversary from unauthorized data access. This happens because any arbitrary sensor is authorized to act on behalf of the user.

1.3 Contributions

The contributions are twofold:

1. This work systematically investigates the problem of authenticated queries in sensor networks in presence of node captures. To the best of my knowledge, such systematical approach was not considered previously. Moreover, solutions to certain problem instances in the literature are very scarce (see Section 7). A precise problem statement is given, and then the design space for solutions is specified.
2. For each possibility in the design space, solutions are discussed. Some of them are already known mechanisms for securing communication networks. For some other cases, original solutions are outlined. However, as this is a work-in-progress report, presented schemes are not fully implemented and analyzed yet.

Roadmap: Section 2 defines authenticated querying and gives design space for its implementation. Section 3 discusses general techniques for query authentication in sensor networks. In Sections 4 and 5, each possibility in design space is considered, and existing solutions are outlined. New solutions are presented in Section 6. Section 7 discusses related work. Section 8 summarizes and describes future work.

¹ I am not aware of such a mechanism for arbitrary sinks, although in [24], secure and authenticated variant of Directed Diffusion for a dedicated sink (the base station) is presented.

2 Problem Statement

2.1 System Model

Sensor network architecture. Consider a sensor network which is deployed over a large geographic area. The network consists of a large number of resource constrained sensor nodes such as MICA2 sensors [5] which have 128 KB flash instruction memory, 4 KB SRAM, an 8-bit microprocessor, and are powered by two AA batteries. The sensor nodes are not tamper proof, i. e., they are susceptible to node capture attacks.

There is also a small number of base stations which have more resources than the sensor nodes. For example, they can be laptop class devices.

Users. The maintainer of the sensor network offers services to a large number of mobile users. Legitimate users can access the sensor network using some mobile device like a PDA or a mobile phone.

Queries. Queries can be injected into the sensor network either at a base station (like in TinyDB [14] or Cougar [27]) or at any sensor node (like in Directed Diffusion [12]). The queries may be first optimized or otherwise processed at the place of injection and then they are disseminated in the sensor network using multihop communication according to some query processing mechanism.

Adversary. The goal of the adversary is to post arbitrary unauthorized queries to the sensor network. The adversary can capture a small amount of sensor nodes, read out their memory contents, and make them run arbitrary programs.

2.2 Authenticated Querying

As the adversary can capture some sensor nodes, he would have access to all data measured by these sensor nodes, and to all data routed through them (in case the data are unprotected or can be decrypted by means of captured cryptographic keys). This data disclosure cannot be prevented in face of node captures. Nevertheless, the adversary should not be able to post *arbitrary* queries to the sensor network.

Definition 1 (Authenticated Querying). *Let WSN be a sensor network consisting of N nodes s_1, \dots, s_N . The users can post queries $q \in Q$ to the WSN. Consider an arbitrary query q . Let S_q be the set of all sensors which must process the query in order to give the required answer to the user. The WSN satisfies authenticated querying if it satisfies the following properties:*

- (Safety) *If a sensor s processes the query q , then q was posted by a legitimate user U .*
- (Liveness) *Any query q posted by a legitimate user U will be processed at least by all sensors $s \in S_q$.*

2.3 Design Space for Authenticated Querying

Two following dimensions can be identified for authenticated querying:

- The user has to communicate with the base station in order to post queries vs. the query can be started at some sensor nodes.
- The sensor network has to forward some data using multihop communication before the user can start posting queries vs. the query can be started locally.

These two dimensions give four possibilities for authenticated querying (AQ) (Table 1): direct base station AQ, remote base station AQ, distributed local AQ, and distributed remote AQ.

Table 1. Design space for realizing authenticated querying (AQ) in sensor networks. “Base station: yes” means that the base station must be accessed before the query processing can be started by the network. “Routing: yes” means that multihop communication is needed before the query processing can be started by the network.

	routing: no	routing: yes
base station: yes	direct base station AQ	routed base station AQ
base station: no	distributed local AQ	distributed remote AQ

Each of these mechanisms is appropriate in different situations and requires different solutions. In the following Section 3, general techniques for query authentication are discussed. In Sections 4 and 5, each mechanism from the design space is considered, and existing solutions are outlined. New solutions are proposed in Section 6.

3 Techniques for Authenticated Querying

Any of following techniques can be used with any mechanism from the design space. However, for clarity of presentation, I do not include them into Table 1, but list them separately.

3.1 Authenticated Broadcast

One possibility for an entity to authenticate its queries (or more generally, messages) is *authenticated broadcast*. This means that one sender can send a message to multiple receivers (here, the sensor nodes). The receivers can verify the origin of the message using some authentication information attached to it.

Some approaches to authenticated broadcast in sensor networks exist in the literature. In SPINS [18], authenticated streaming broadcast μ TESLA is realized using one-way hash chains, time synchronization, and symmetric keys shared by the base station with each sensor in the network.

Inexpensive digital signatures can also be used for authenticated broadcast (see e.g. [21]), assuming that each sensor node is preloaded with the public key of some certification authority. For discussion of public key cryptography in sensor networks, see Appendix A.

However, symmetric key cryptography should be preferred in sensor networks. Lower bounds on authenticated broadcast which uses only message authentication codes (MACs) are considered by Boneh et al. in [3]. Scheme for authenticated broadcast of Canetti et al. [4] meets this lower bound. It requires over 800 bits of authentication information per message, assuming that up to 10 receivers can collude and forge an authenticated message for a particular receiver with probability 2^{-10} . This scheme is independent from the number of receivers. Such high communication overhead is prohibitive in sensor networks. Below I argue that nevertheless, purely symmetric techniques can be considered for sensor networks.

3.2 Authenticated Flooding and Cooperative Approaches

Broadcast authentication protocol μ TESLA achieves much better performance than the lower bound showed in [3]. This happens because, additionally to the system model used by Boneh et al., time synchronization is assumed between the sender and the receivers.

In sensor networks, another (implicit) assumption from [3] does not necessarily hold. This assumption states that the receivers do not communicate or cooperate with each other. However, in sensor networks, most queries are forwarded over multihop communication, or even flooded. Lower bounds from [3] do not apply if cooperation between the receivers is assumed. Therefore, more efficient methods which use symmetric cryptography may be possible. For an example of authenticated flooding, see Section 6.1.

An example of cooperative approach is interleaved message authentication from [22, 30]. There, sensor nodes on the multihop route along which a message is forwarded, cooperatively authenticate this message. This helps to withstand capture of some fixed number of sensor nodes. See Section 6.2 for an outline of an authenticated querying algorithm which uses interleaved message authentication.

4 Using Base Stations for Authenticated Querying

Base stations are supposed to have more resources and to be better protected against attacks than sensor nodes. Therefore, using base station is a natural approach to organize such a critical task as authenticated querying.

4.1 Direct Base Station Authenticated Querying

The query is always started at the base station, either by physically approaching it with a device and connecting to it (wirelessly or not), or by routing the query through some external network (e.g., the Internet) which is connected to the base station. Users log into the base station using an arbitrary client/server

authentication protocol [16]. If the user is successfully authenticated, he can post arbitrary queries to the base station. The base station forwards (possibly optimized) user's queries into the sensor network.

In this case, the base station also has to authenticate its query such that all sensors which process the query can verify that it originated at the base station. However, the base station is a trusted entity, and therefore, has more opportunities for query authentication than a user. For instance, it may know the network topology, know at which nodes the data are stored, and share symmetric keys with each sensor in the network.

4.2 Remote Base Station Authenticated Querying

The query can be started on some sensor node (or nodes). Sensors are not concerned with query authentication, but just route the authentication information to the base station, the base station authenticates the user and then gives permission to the sensor network to answer user's queries. Thus, the base station helps to establish trust between the sensor network and the users. Here, at least two scenarios are possible.

(1) User's queries are always routed to the base station. The base station sends authenticated queries into the network on behalf of the user, receives the answers, and sends the answers back to the user. In this case, an SSL-like protocol can be used to set up a secure authenticated channel between the user and the base station.

(2) The base station generates a kind of "ticket" (using Kerberos terminology) which enables the user to talk to the sensor network for some time. The ticket is sent back to the user who uses it to generate authenticated queries. For a possible solution, see Section 6.1.

4.3 Authenticated Querying Using Base Station: Pros and Cons

(Advantages) The base station has more resources than a sensor and therefore, can implement stronger security measures. It can be placed in dedicated locations and maintained by humans. This makes the base station more reliable and secure: it can be protected from physical access, and DoS or penetration attacks are more likely to be spotted quickly.

(Disadvantages). The base station serves as a dedicated authentication server. Therefore, it must be very well protected from both physical and remote access by unauthorized entities. In the literature is usually assumed that to take over a base station is more difficult than a sensor node. In practice, the reverse could be the case. For example, if the base station is connected to a popular web server with known vulnerabilities, the penetration of the base station could be a matter of utilizing an available exploit. Besides, it is not always possible or desirable to place a base station into a dedicated secure location, especially if it is supposed to communicate with the sensors wirelessly.

Furthermore, if the direct physical access is needed for the authentication (e.g., wireless communication with the base station), it might be inconvenient to the user to walk through the half deployment area to the base station while needing the data from sensors in user's proximity. In case of remote access, several messages have to be routed between the base station and the user by the sensor network, which can be impractical if the base station is far away. The user might also have to wait for the answer of the far away base station while needing the data from sensors close-by.

And finally, as the sensors close to the base station are more heavily loaded with communication, their energy is exhausted more quickly, which leads to shorter network lifetime.

5 Authenticated Querying Without Base Stations

In cases where the base station cannot be used for authenticated queries, an access control mechanism or, more generally, a mechanism for query verification, should be built into sensor nodes. However, as shown in Section 1.2, relying on any arbitrary sensor for access control is not sound in face of node capture attacks. A natural solution here would be to use some kind of distributed algorithm, therefore the word “distributed” for the case without base station.

5.1 Distributed Local Authenticated Querying

The legitimacy of the query is verified by the sensors in user's location, e.g., in his communication range. Of course, even if the sensors in user's proximity successfully verified the query, they still need some means to tell to the rest of the sensor network that this query comes from a legitimate user. That is, at least sensors which process the query should be able to verify its legitimacy. For a concrete proposal, see Section 6.2.

5.2 Distributed Remote Authenticated Querying

The legitimacy of the query is verified by several sensors. These sensors can be specially chosen for this purpose, then the network architecture might be heterogeneous, with dedicated authentication devices placed in some locations. On the other hand, these sensors might be selected from the set of all sensors according to some algorithm.

Distributed remote authenticated querying can also be organized if each sensor in the network can verify the legitimacy of the query. For example, each user could receive a private/public key pair, certified by the certification authority of the maintainer, for signing his queries. Each sensor in this case must be preloaded with the authentic copy of the public key of the certification authority. Each query is then digitally signed and sent into the sensor network together with the corresponding user's certificate. In order to reduce computational burden on the sensor nodes, each sensor node could decide whether it is to verify the query with some probability.

According to Appendix A, signature verification can be an efficient operation. However, overhead for verifying user-generated signatures is twice as large as for signatures generated by the base station (see Appendix A.3).

5.3 Authenticated Querying Without Base Station: Pros and Cons

(Advantages) Users can start queries locally in the sensor network, without going to the base station or some other access point (e.g., an Internet terminal). Furthermore, the query can be processed and answered locally. No routing to the base station is needed for answering the query. Still, routing can be needed if the query concerns sensor data which are not in the user's proximity. And last but not least, if the base station is overloaded or taken over, the data are still available and not compromised (at least, as long as the compromised base station can be excluded from the network management).

(Disadvantages) The most severe disadvantage is that user authentication costs extra computation and communication power, especially if it is done in distributed fashion, using replication and agreement techniques, or public key cryptography. Distributed algorithms have to be applied in order to cope with unreliability and insecurity of individual nodes.

6 New Ideas for Authenticated Querying

Here, I present new proposals and ongoing work on authenticated querying.

6.1 Ticket Generation for Remote Base Station Authenticated Querying

I propose ticket generation using ID-based key predistribution. Key predistribution for sensor networks originates from [8]. The idea is that each sensor node is preloaded with randomly chosen m keys from the key pool of size l . The values of l and m can be chosen such that any two nodes have at least one common key with a given probability. ID-based key predistribution was introduced in [31]. The keys in the key pool are numbered from 1 to l . Each sensor with a unique identifier id is first assigned m distinct integers between 1 and l by applying a pseudo random number generator $PRG()$ with seed id . This method of choosing keys enables any sensor node u which knows the identifier id_v of another sensor node v to compute v 's key identifiers by computing $PRG(id_v)$ and thus determine if u and v share some keys.

I adapt the above scheme to authenticated querying. In my idea, keys (and their identifiers) are predistributed to the sensor nodes using $PRG(id_{sec})$ where id_{sec} are *secret* sensor identifiers known only to the base station, but not stored on the sensor nodes. These identifiers should be sufficiently large (e.g., 80 bits) to prohibit the brute-force search of an identifier which generated a particular set of key identifiers. Therefore, an adversary who captured a sensor node cannot determine the secret identifier id_{sec} from which key identifiers were derived.

If a user U successfully authenticated to the base station, he receives from the base station a temporal identifier id_U (with a time stamp) and a set K_U of m secret keys which the base station computed using $PRG(id_U)$. Now the user formulates his query q and computes $h(q)$ where h is a hash function. After that, the user computes m 1-bit message authentication codes (MACs) on $h(q)$ using keys from K_U . The idea of using MACs with single bit output originates from [4].

The query accompanied by m 1-bit MACs and user's temporal identifier id_U is sent into the sensor network. Each sensor s can compute $PRG(id_U)$ and thus determine whether some of 1-bit MACs were computed with keys known to it. Then s verifies all 1-bit MACs which it is able to verify. If any of them is wrong, the node discards the query. Otherwise, it forwards the query to its neighbors.

This scheme is an example of authenticated flooding (see Section 3.2). The query of a legitimate user will be flooded into the sensor network without any obstacles. However, a query forged by an adversary will only be able to reach a limited part of the network, as some sensor nodes will discard the query. An initial analysis of query propagation in this model shows that a relatively small number of 1-bit MACs suffices to limit the query propagation to a logarithmically small part of the network. Thus, if there are N sensor nodes, approximately $\ln N$ sensor nodes will receive a fake query. For example, for $N = 10,000$, 300 1-bit MACs suffice. Further analysis of this scheme is subject to ongoing work.

The communication overhead of this scheme is rather high. If we assume that each query is accompanied by 300 bits of MACs and a user identifier which is 80 bits long, this results in 380 bits, or 48 bytes, of authentication information. The payload of a TinyOS² message is 29 bytes. Thus, each query is accompanied by two packets of authentication information. However, in contrast, an RSA-1024 digital signature which provides very strong message authentication (each sensor can verify the legitimacy of each message) is 1024 bits long, and therefore, requires five TinyOS packets. See Appendixes A.2 and A.3 for more information about using digital signatures in sensor networks.

6.2 Distributed Local AQ

If the number of users is large, the natural method to use for authentication is public key cryptography because of its scalability. On the other hand, public key cryptography is power-hungry, so the sensors should communicate with each other using symmetric cryptography. My concept is to let the sensors in the communication range of the user serve as interpreter (or a gateway) between the "public key crypto world" of the user and the "symmetric crypto world" of the WSN. The user talks to sensors in his communication range using public key cryptography, and these sensors then talk to the remainder of the sensor network on behalf of the user using symmetric cryptography. This happens in authenticated fashion:

² TinyOS is a popular operating system for sensor networks, see e.g. [13].

1. *Robust secure channel setup between the user and the WSN*: The user executes a mutually authenticated key establishment protocol [16] using public key cryptography with a specified number m of sensors in his communication range. The protocol results in establishment of symmetric session keys between the user and each correct sensor which participated in a protocol run.
2. *Authenticated query forwarding*: After the successful secure channel setup, the sensors in user's proximity forward user's query into the sensor network and append to it some additional information which enables the other sensors to verify the legitimacy of the query. In this case, the other sensors should be able to verify that at least m sensors approved the query.

We partially implemented the first step on Telos Revision B sensor nodes [17] using the EccM library for elliptic curve cryptography (ECC) [15]. For details, see [1]. In our implementation, the user unilaterally authenticates to the sensors in his proximity using public key cryptography. Moreover, a great breakthrough was recently reported by Gupta et al. [10]. They implemented the SSL protocol on MICA2 motes using elliptic curve cryptography. This confirms the choice of ECC for our implementation and the feasibility of mutually authenticated key establishment on sensor nodes, as their SSL handshake takes less than 4 seconds.

One possible solution to the second step (query forwarding) would be to use interleaved message authentication [22, 30] with $m - 1$ as a parameter for the number of captured sensor nodes. This is our ongoing work.

7 Related Work

In SPINS [18], authenticated streaming broadcast μ TESLA is realized using one-way hash chains and time synchronization. In [24], one-way hash chains help to authenticate queries in Directed Diffusion. Inexpensive digital signatures are used in [21].

LEAP [29] and LKHW [19] consider using a single symmetric key for authenticated querying and therefore, are not resistant against impersonation attacks which are possible in case of node captures. Hierarchical sensor network architecture is considered in [6], [7]. This helps to support in-network processing, but the cluster heads become very attractive targets for node capture attacks.

All above approaches are suitable for sensor networks with the large number of users only in case of direct base station authenticated querying. To the best of my knowledge, the problem of enabling a large number of mobile users to post authenticated queries to a sensor network was not considered previously.

There is also a number of papers where the "opposite direction" is considered, e.g., verification of the legitimacy and correctness of answers to the queries. These methods cannot be directly applied to authenticated querying, as they assume that the verifier (the base station) has more resources than the sensor nodes. Statistical approaches are considered in [20, 23, 28]. Using clustered sensor network architecture and symmetric key techniques is described in [21, 30].

8 Conclusions and Future Work

I defined the problem of *authenticated querying*, systematically examined the design space for its realizing, considered existing solutions and proposed some new methods for authenticated querying. These are first results of an ongoing project on access control to sensor network data. There is still a lot of work to be done. Proposed solutions have to be precisely specified. Security and resource demands of all solutions have to be more detailed analyzed, theoretically as well as by simulations and by implementation on real sensor nodes. First steps towards these goals are made in [2, 1].

References

1. Z. Benenson, N. Gedicke, and O. Raivio. Realizing robust user authentication in sensor networks. In *Real-World Wireless Sensor Networks (REALWSN)*, Stockholm, June 2005.
2. Z. Benenson, F. C. Gärtner, and D. Kesdogan. An algorithmic framework for robust access control in wireless sensor networks. In *Second European Workshop on Wireless Sensor Networks (EWSN)*, January 2005.
3. D. Boneh, G. Durfee, and M. K. Franklin. Lower bounds for multicast message authentication. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 437–452. Springer-Verlag, 2001.
4. 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*, volume 2, pages 708–716, New York, NY, Mar. 1999. IEEE.
5. Crossbow, Inc. MICA2 data sheet. Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2n_Datasheet.pdf.
6. J. Deng, R. Han, and S. Mishra. Security support for in-network processing in wireless sensor networks. In *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 83–93, New York, NY, USA, 2003. ACM Press.
7. T. Dimitriou and D. Foteinakis. Secure and efficient in-network processing for sensor networks. In *First Workshop on Broadband Advanced Sensor Networks (BaseNets)*, 2004.
8. L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM Press, 2002.
9. G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In *ESAS*, pages 2–18, 2004.
10. V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, and S. C. Shantz. Sizze: A standards-based end-to-end security architecture for the embedded internet. In *Third IEEE International Conference on Pervasive Computing and Communication (PerCom 2005)*, Kauai, March 2005.
11. N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *CHES2004*, volume 3156 of *LNCS*, 2004.

12. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed Diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.
13. P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler. The emergence of networking abstractions and techniques in tinyos. In *First USENIX/ACM Symposium on Networked Systems Design and Implementation*, 2004.
14. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 491–502, New York, NY, USA, 2003. ACM Press.
15. D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *First IEEE International Conference on Sensor and Ad Hoc Communications and Network*, Santa Clara, California, October 2004.
16. A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 1997.
17. Moteiv, Inc. Tmote Sky datasheet. Available at <http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf>.
18. A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: security protocols for sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 189–199. ACM Press, 2001.
19. R. D. Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. J. M. Havinga. LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks. In *32nd International Conference on Parallel Processing Workshops (ICPP 2003 Workshops)*, 2003.
20. B. Przydatek, D. Song, and A. Perrig. SIA: Secure information aggregation in sensor networks. In *ACM SenSys 2003*, Nov 2003.
21. S. Seys and B. Preneel. Efficient cooperative signatures: A novel authentication scheme for sensor networks. In *2nd International Conference on Security in Pervasive Computing*, number 3450 in LNCS, pages 86 – 100, April 2005.
22. H. Vogt. Exploring message authentication in sensor networks. In *Security in Ad-hoc and Sensor Networks (ESAS), First European Workshop*, volume 3313 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2004.
23. D. Wagner. Resilient aggregation in sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 78–87. ACM Press, 2004.
24. X. Wang, L. Yang, and K. Chen. Sdd: Secure distributed diffusion protocol for sensor networks. In *First European Workshop on Security in Ad-hoc and Sensor Networks(ESAS)*, volume 3313 of *Lecture Notes in Computer Science*, pages 205–214, 2004.
25. R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus. TinyPK: securing sensor networks with public key technology. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 59–64. ACM Press, 2004.
26. A. Woo, S. Madden, and R. Govindan. Networking support for query processing in sensor networks. *Commun. ACM*, 47(6):47–52, 2004.
27. Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, 31(3):9–18, 2002.
28. F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route detection and filtering of injected false data in sensor networks. In *Proceedings of IEEE INFOCOM*, 2004.

29. S. Zhu, S. Setia, and S. Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 62–72. ACM Press, 2003.
30. S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *IEEE Symposium on Security and Privacy*, pages 259–271, 2004.
31. S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach. In *IEEE International Conference on Network Protocols*, November 2003.

A Asymmetric Key Cryptography in Sensor Networks

A.1 Efficiency Considerations

Although symmetric key cryptography is several orders of magnitude more efficient than asymmetric key cryptography [16], quite a number of researchers considered implementing efficient public key cryptography on small devices in the last few years. The reasons are that asymmetric key cryptosystems scale much better and allow more flexible key management, e.g., key agreement. Moreover, some asymmetric key cryptosystems allow efficient algorithms for encryption and for verification of digital signatures. In contrast, digital signature schemes based on symmetric mechanisms usually require large verification keys (cf. [16, page 31]).

A.2 Choosing Appropriate Asymmetric Key Cryptosystem

RSA with small public exponent (considered in [11, 25]) and Rabin public key cryptosystems (considered in [9]) have fast algorithms for encryption and digital signature verification. However, decryption and signature generation are slow and resource-demanding. Therefore, these cryptosystems can be used in sensor networks only if the sensors are not required to decrypt or to sign messages.

In contrast, elliptic curve cryptosystems (ECC, considered in [15]) require more overhead for encryption and signature verification than for decryption and signing. Nevertheless, with ECC, not only encryption and signature verification, but also decryption and signing are feasible for sensor nodes. Recently, Gupta et al. [10] implemented a very efficient SSL-like protocol on MICA2 sensor nodes using ECC in assembly language.

A.3 Digital Signatures by Base Stations vs. by Users

In some algorithms discussed in the main part of this paper, sensor nodes have to verify digitally signed messages originated from the base station or from the users. Each sensor has an authentic copy of base station’s public key preloaded. To verify signatures of the base station, a sensor node just needs to apply the preloaded public key of the base station to the signature.

To be able send digitally signed messages to the sensor nodes, each user receives from the base station a certificate. This certificate is essentially user's public key signed by the base station. To verify user's signature of a message, the sensor network needs first to verify user's certificate. Thus, in case of user-signed messages, a sensor node needs to verify two signatures: First, the base station's signature on the certificate, and second, user's signature on the message. This means that verifying user-generated signatures roughly requires twice more resources than verifying signatures generated by the base station.