User Authentication in Sensor Networks (Extended Abstract)

Zinaida Benenson Felix Gärtner Dogan Kesdogan

RWTH Aachen, Department of Computer Science, D-52056 Aachen, Germany

Abstract: If the data collected within a sensor network is valuable or should be kept confidential then security measures should protect the access to this data. We focus on user authentication, a central problem when trying to build access control mechanisms for sensor networks. We first sketch some security issues in the context of user authentication in sensor networks. We then introduce the notion of n-authentication, a special form of authentication which is more adequate to sensor networks than previous forms of authentication. We finally present and analyze a protocol for n-authentication.

1 Introduction

Wireless Sensor Networks (WSNs) are networks of tiny sensing devices which are spread over a large geographic area and can be used to collect and process environmental data like temperature, humidity, light conditions, seismic activities, images of the environment etc. This data can be used to detect certain events and to trigger activities. For example, sensors distributed over a large woodland could automatically raise an alarm if a fire has broken out somewhere, or sensors distributed over a large farmland could trigger irrigation if the ground of a field is not moist enough.

With the increasing ubiquity of WSNs, environmental data will be available almost everywhere in our environment. We believe that in the future the current temperature, humidity, etc. at a particular location will be available on demand from a surrounding WSN. Of course, accessing this data will in general not be for free since deployment of WSNs induces some costs. This means that the deployment agencies of some of these services will make them available only to "authorized" people (i.e., paying customers). In this case, a WSN must be able to distinguish legitimate users from illegitimate users, resulting in the problem of access control.

Access control is an old problem from classical computer science but has not received much attention in the context of WSNs. This is unfortunate since WSNs define an environment which naturally calls for security solutions but — due to the resource-constraints with respect to computational and battery power for example — also defines an environment in which security solutions are extremely hard to implement.

This extended abstract investigates the problem of access control in WSNs. More specifi-

cally, we focus on the problem of *user authentication* in WSNs, an important subproblem of access control. We give a brief overview over the security issues in the context of protecting sensor network data (Section 2). In Section 3 we then introduce the notion of n-authentication, a special form of authentication which is more adequate to sensor networks than previous forms of authentication and present a protocol for n-authentication. We analyze and discuss the protocol and its assumptions in Section 4 and conclude in Section 5. For lack of space, many of the details in this abstract are deferred to the extended version of this paper which will be published later.

2 Security Issues in WSNs

If the data collected within a sensor network is valuable or should be kept confidential then we need to control the access to this data in convenient ways. To separate concerns we propose to distinguish *inside security* and *outside security* for WSNs.

Inside security refers to secure communication between the sensors and secure communication between the base stations (if there are any) and the sensors. In this case, base stations are usually considered to be trusted and to have a similar authorization as network administrators in classical networks. Among the security problems evolving in WSNs, inside security has been studied most extensively [PSW+01, CPS03, ZSJ03].

Outside security means secure communication between the WSN (sensors and base stations) and the outside users, i.e., the "subscribers" to WSN services. A legitimate user can send data requests to the WSN. Usually it means that the user sends the request to some sensor or a set of sensors in her neighborhood and — if the request is legitimate — receives a valid response.

Data integrity and availability are important aspects of outside security and have been studied, e.g., in [PSP03] and [WS02]. On the other hand, access control, the heart of solutions to confidentiality (and integrity) has not received much attention yet. The most important part of any access control solution is *user authentication*.

3 User Authentication in WSNs

3.1 Simple Authentication

Menezes et al. [MOV97, p. 386] define the term *entity authentication* as "...the process whereby one party is assured [...] of the identity of a second party involved in a protocol...". We call the two players involved prover P and verifier V. The verifier is requested by the prover to establish a correct relation between a particular identity and the prover.

There can be multiple provers having the same identity, e.g., Alice's PDA, her workstation or her mobile phone can all be associated with the identity of Alice. We assume that a prover has at most one identity. We denote the set of all identities by $\mathcal{I}$.

We now formally define the properties of authentication protocols. These properties are defined with respect to the two primitive operations of authentication: (1) *authenticate*(V, I) is invoked by the prover P whenever P would like to be authenticated by V using identity $I \in \mathcal{I}$; (2) *associate*(P, I) is invoked by the verifier whenever it has established the relation between P and some identity I. Intuitively, an authentication protocol is correct if the identity associated to P by V is the "real" identity of P. If P is dishonest or claims to have a fake identity this is indicated by a special value $\perp$ which is supposed to be distinct from any value in $\mathcal{I}$. Authentication is *successful* if V invokes *associate*(P, I) with some $I \neq \perp$.

More precisely, a protocol solves authentication if it guarantees two properties:

- (Validity) An honest verifier V invokes *associate*(P, I) with $I \in \mathcal{I}$ only if P in fact has identity I.
- (Termination) If P invokes *authenticate*(V, I) and if V is honest then V will eventually invoke *associate*(P, I') for some identity $I' \in \mathcal{I}$ or $I' = \perp$.

We call a protocol which satisfies the above two conditions a *simple authentication protocol*. Simple authentication is not sufficient in wireless sensor networks if failures and active adversaries are taken into account. If we require that a prover (i.e., a user) always authenticates to some particular sensor, then this becomes impossible if that sensor fails. However, if we don't care which sensor the prover uses for authentication, then taking control of a single sensor is sufficient for an active adversary to gain access to the entire system. What is needed is a more robust notion of authentication.

3.2 n-Authentication

We now introduce the notion of n-authentication, a robust version of simple authentication. To be robust against failures, this new form of authentication succeeds if the user can successfully authenticate with any subset of sensors out of a set of n sensors (n can be the average number of sensors within broadcast distance of the user). To be robust against active attacks where the adversary can compromise up to t sensors ($t < n$), we require that the subset of sensors to which the prover has to authenticate has at least the size of $n - t$.

More formally, we now consider a set of n verifiers $\mathcal{V} = \{V_1, \ldots, V_n\}$. To distinguish the primitive operations of simple authentication from those of n-authentication we denote the latter ones with n-*associate*(P, I) and n-*authenticate*$(\mathcal{V}, I)$. Note that n-authenticate refers to the entire set of verifiers while n-associate just refers to a single prover.

A protocol solves n-authentication if it satisfies the following properties:

- (Termination) If P invokes n-*authenticate*$(\mathcal{V}, I)$ then eventually all honest $V_i \in \mathcal{V}$ invoke n-*associate*(P, I_i) for some $I_i \in \mathcal{I}$ or $I = \perp$.
- (Validity) An honest verifier V_i invokes n-*associate*(P, I) only if P in fact has identity $I \in \mathcal{I}$.

- (Agreement) If honest verifier V_i invokes n-*associate*(P, I') and honest verifier V_j invokes n-*associate*(P, I'') then $I' = I''$.

If we assume that at most t verifiers fail, then n-authentication ensures that the remaining (at least $n-t$) verifiers eventually successfully authenticate an honest prover and that they agree on its identity. If a prover is dishonest or claims to have a fake identity then all honest verifiers will return $\perp$ so that the prover is not authenticated.

3.3 Implementing n-Authentication

We now describe a generic protocol for n-authentication in WSNs. It builds upon a protocol for simple authentication. The full version of this paper will present different solutions based either on symmetric key cryptography or on zero knowledge proofs and discuss their merits.

Consider a user P approaching a WSN. Let n sensors $V_1, \ldots, V_n$ be in the communication range of P. We assume that inside security guarantees authenticity, integrity, confidentiality and freshness of messages sent between the sensors. Furthermore, we assume that $V_1, \ldots, V_n$ are in communication range of each other. We discuss adequacy of this assumption in the full version of this paper. The approach works as follows:

1. P authenticates separately to each of nodes $V_1, \ldots, V_n$ using a method for simple authentication. Here the communication between P and all n sensors must be organized without any collision, which is one of the main challenges here. For example, P can coordinate the communication by means of a TDMA schedule.
2. If P successfully authenticated itself to a node V_i, then V_i broadcasts to the other nodes its vote *yes*. Otherwise, V_i sends nothing.
3. Each sensor V_i sets a timeout, collects the votes and successfully authenticates P only if $n-t$ or more *yes*-votes are collected. Otherwise, i.e., if $t+1$ or more votes fail to be received before the sensor times out, the authentication is unsuccessful.

4 Discussion and Analysis

Correctness. The protocol will terminate if either at least $n-t$ *yes* votes were collected (successful authentication) or if the sensor times out after receiving the initial user request (unsuccessful authentication). The validity property of n-authentication is guaranteed by the validity property of the solution to simple authentication used in step 1. Finally, agreement is ensured by the properties of the secure broadcast channels (following from inside security), and the assumption that all n sensors are within each other's broadcast range. In this case, if a honest sensor V_i successfully n-authenticates P, then V_i received at least $n-t$ *yes* votes. Consequently, all other honest sensors also receive these votes. The case

of unsuccessful authentication can be argued similarly. In order for this scheme to give unique results it is necessary that $t < n/2$, i.e., it requires a majority of honest sensors.

Communication Efficiency. Requiring the broadcast of a vote by every sensor during step 2 of the protocol imposes a large communication overhead which may be prohibitive in resource constrained networks. If P has to authenticate to each verifier V_i separately using different authentication information, we conjecture that the lower bound on the number of messages is indeed $\Omega(n)$ as in our protocol. On the other hand, if P can authenticate with the same information by all verifiers $V_1, \ldots, V_n$, protocols with $O(1)$ messages are possible. However, we doubt that these protocols can be of any practical relevance. We investigate this issue in the full version of this paper.

5 Conclusions

We have sketched some security issues in the context of user authentication in WSNs. We have introduced the notion of n-authentication, which is more adequate to WSNs than simple authentication, and have given and analyzed a protocol for n-authentication.

References

[CPS03] Chan, H., Perrig, A., und Song, D.: Random key predistribution schemes for sensor networks. In: *IEEE Symposium on Security and Privacy*. S. 197–213. May 2003.

[MOV97] Menezes, A. J., Oorschot, P. C. V., und Vanstone, S. A.: *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL. 1997.

[PSP03] Przydatek, B., Song, D., und Perrig, A.: SIA: Secure information aggregation in sensor networks. In: *ACM SenSys 2003*. Nov 2003.

[PSW+01] Perrig, A., Szewczyk, R., Wen, V., Culler, D., und Tygar, J. D.: Spins: security protocols for sensor netowrks. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. S. 189–199. ACM Press. 2001.

[WS02] Wood, A. D. und Stankovic, J. A.: Denial of service in sensor networks. *Computer*. 35(10):54–62. 2002.

[ZSJ03] Zhu, S., Setia, S., und Jajodia, S.: Leap: efficient security mechanisms for large-scale distributed sensor networks. In: *Proceedings of the 10th ACM conference on Computer and communication security*. S. 62–72. ACM Press. 2003.