Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Integrity Preservation for Communication in Sensor Networks

Technical Report 434

Department of Computer Science
Institute for Pervasive Computing
ETH Zürich

Harald Vogt

Integrity Preservation for Communication in Sensor Networks

Harald Vogt
Institute for Pervasive Computing
ETH Zürich, Switzerland
vogt@inf.ethz.ch

February 17, 2004

Abstract

We propose a novel data integrity protection scheme, which relies on multiple, intervoven authentication chains instead of data origin authentication. We show that the scheme allows for secure node-to-node communication at very low implementation cost, under a certain attacker model, without reliance on a base station.

1 Introduction

In a wireless sensor network (WSN), a large number of small, resource-restricted nodes cooperate to achieve a monitoring task. A common application scenario for WSNs is the tracking of moving objects through a geographical area where a sensor network is deployed. Individual sensor findings are aggregated and fused, possibly in a distributed manner, to infer higher-level information about the environment of the sensor network. Algorithms for WSNs should be designed such that a certain amount of failed nodes can be tolerated without compromising the reliablility of results. However, this is not sufficient if a WSN is under attack from an adversary that tries to induce false data into the network, since compromised nodes, in contrast to failed nodes, cannot always be identified.

If a sensor node is taken over (and left intact) by the attacker, the attacker can induce arbitrary messages into the network originating from this node (e.g., by manipulating its sensoric input). There is no way of hindering the attacker from doing so. However, an effective security scheme must detain the attacker from inducing messages that look as if they were originating from *another* node (or, equivalently, changing the contents of such messages). This is conventionally achieved by authenticating individual messages with a key only known to the sender and the recipient of the message.

Instead of overtaking existing nodes, the attacker can deploy his own sensors and try to fool the network into accepting messages from them. This can be prevented if truthful nodes are required to hold a certificate or a shared key to authenticate themselves to other nodes. Additionally, deployed sensor nodes must be sufficiently protected against (physical) tampering to prevent the extraction of key material.

In this paper, we present a scheme that complements existing key distribution schemes for WSNs and protects the communication within a WSN against an attacker who tries to manipulate messages in the network. The scheme relies on symmetric cryptography, taking the restrictions of sensor nodes into account. It abstains from using public-key cryptography or a complete infrastructure of mutually shared symmetric keys, and does not require a base station. Nevertheless, it allows for reliable communication among any pair of nodes. We call the scheme *Canvas*.

1.1 Related Work

The use use of multiple transmission paths for increasing the reliability of communications is a well-known concept [3, 8, 5]. We are not aware of any work that has considered the concept of interwoven authentication paths, using a single data transmission path, though. As our early results indicate (see Sect. 4), interweaving can raise the demands on an attacker.

A similar idea as used in *Canvas* has been investigated in [2] for the protection of results that a roaming agent has gathered visiting several hosts. Before an agent leaves a host, the predecessing host is required to co-sign the agent's new result list. This prevents the host from deleting intermediary results in the result chain (and thereby giving its own result higher priority). In this approach, the number of co-signing hosts can be increased for protection against more colluding attackers. This is in contrast to *Canvas*, where nodes farther away from each other protect against more compromised nodes.

The μTESLA protocol [9] for sensor networks achieves authentication for sensor nodes communicating with a base station, based on symmetric cryptography and self-authenticating key chains. The main difference to our approach is the fact that all communication must pass through the base station. However, one also gets origin authentication for messages, not only data integrity.

2 Data Integrity in Sensor Networks

In a sensor network, a vast number of objects with identical physical appearance and identical behaviour is deployed. In most sensor network applications, a monitored phenomenon is determined upon the basis of a collection of various sensor findings. The sensors are cooperating in performing a computation, or they are sending their findings to a base station, where the data is evaluated. In this process, it is not of great importance from *which* sensor instance the data are originially. Other factors are more important, such as geographical position. (The geographical "identity" of a sensor could be assured as described in [6, 10].)

It is, however, also important to preserve accuracy and integrity of the sensor findings. Whereas accuracy is guaranteed through carefully engineering the sensor hardware and software, integrity is conventionally achieved through *data origin authentication*. Data origin authentication requires that each data source have an identity on which the authentication is based. A prerequisite for data origin authentication is the availability of either public-key cryptography or mutually shared symmetric keys. Both approaches have their limitations in sensor networks. Public-key cryptography is often considered too computationally intensive for small sensor nodes. With symmetric keys, if communication between any two nodes must be possible, each node would have to store $O(n)$ keys, which soon exceeds the storage capacity of sensor nodes. An alternative approach is presented in [4]. In this approach, randomly pre-distributed key sets are used to establish mutually shared keys for node pairs. Our approach complements this work by making use of these keys for local secure communication amongst neighbouring nodes.

Another problem with data orign authentication arises when the receiver of a message is not known in advance, or when there are many receivers. This problem is especially prevalent in content-based data distribution.

These problems are addressed by the *Canvas* scheme. It provides data integrity without using end-to-end security features. Only local authentication of messages is used. This heavily reduces the requirements on sensor nodes, which have to store only a very small number of keys.

3 The *Canvas* Scheme

The *Canvas* Scheme consists of three phases. The task of the first phase is *key pre-distribution*. It is carried out before the sensor network is deployed. At the end of this phase, an arbitrarily chosen pair of nodes is (with high probability) able to establish a secret shared key (a suitable approach is described in [11]).

The second phase follows immediately after deployment, when the distribution of the sensor nodes has been fixed. (We do not consider mobile nodes in this paper.) Each node establishes a separate secret shared key with each of its immediate (1-hop) and indirect (2-hop) neighbours. We assume that only such nodes can participate in this process, which also participated in the first phase. This prevents an attacker from joining the network with his own nodes.

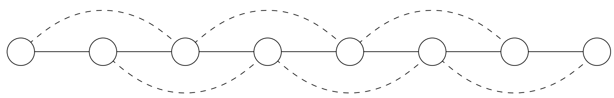

Figure 1: A path between two communication endpoints. The solid lines represent physical links. The dashed lines indicate additional shared keys

After the second phase, there exists at least one path between any two nodes in the network (if the network is connected) with the characteristics shown in Fig 1. Apparently, an attacker can manipulate messages on such a path if he controls two adjacent nodes. Single nodes under the attacker's control are not capable of disrupting the communication path.

The third and last phase is the operational phase of the sensor network. Nodes exchange messages with remote peers by "authenticating" them with their neighbour keys along the transmission path. This will be explained in detail below. Note that we assume a suitable routing scheme.

3.1 Key Pre-Distribution

Before deployment, each sensor is given a piece of data that allows it later to establish a shared key with each other node. In a simple application scenario, this could be a global key, shared amongst all nodes. Since this key is used only in the following key establishment phase (and deleted afterwards), this may be a feasible approach in some application scenarios.

If the nodes are in principle capable of performing public-key cryptographic operations, a solution is to give each sensor node a private key and a certified public key that is only used in the subsequent key agreement phase. The *Canvas* scheme can help to avoid further costly public-key operations during the operational phase.

Another viable approach is probabilistic key agreement as described in [11]. From a large key set, each node is given a small subset. The choice of the subset is determined by a pseudo-randomly generated index set, based on a unique node identifier. This guarantees, with high probability, both that no pair of nodes holds the same set of keys, but also that any two nodes have a certain minimum number of keys in common (and this set of shared keys is unique to that pair).

3.2 Pairwise Key Agreement

Pairwise key agreement depends on the key pre-distribution phase. If a globally shared key is being used, pairwise key agreement is very simple, but not very secure. With this approach, the assumption is necessary that the second phase is being carried out while the network can be considered protected from any influence of the attacker. It must be ensured that the globally shared key is removed from all nodes before the second phase ends. Otherwise, an attacker could extract the global key from a compromised node. He could then reconstruct the local keys from the recorded traffic of the second phase.

With public/private key pairs, it is possible at any time for any two nodes to establish a secret key [7]. When establishing a key with a 2-hop neighbour, it is crucial to avoid a man-in-the-middle attack, which would allow an attacker to pose as two distinct nodes while in fact, there is only one active node (a 1-hop neighbour).

The probabilistic approach is both flexible and robust. Additional nodes can be added to the network later (the key agreement phase can overlap the operational phase). Impersonating a node is only possible if the key set associated with the impersonated identifier is completely known. This is only feasible if the attacker manages to compromise a very large number of nodes and collect their keys. (A comprehensive analysis is omitted here but is subject to future research.) If knowledge of the impersonated key set is incomplete, probability is high that a (truthful) peer will refuse to agree on a shared key. Therefore, it is safe to exchange messages for key agreement with a 2-hop neighbour by using a 1-hop neighbour as a bridge.

3.3 Communication During Operation

Two nodes in a WSN communicate along at least one path of multiple intermediary nodes, determined by a suitable routing algorithm. We believe that *Canvas* is flexible enough to support a large variety of these algorithms. For now we assume that there are two nodes S_1 and S_n, where S_1 wants to transmit a message to S_n. We also assume that there is a communication path $\langle S_1, S_2, \ldots, S_{(n-1)}, S_n \rangle$ (intermediary nodes need not be fully known to neither S_1 nor S_n, also S_1 doesn't have to know about the actual identity of S_n).

In the following, shared keys will be denoted as $K_{i,j}$ for a key shared between nodes S_i and S_j. (For simplicity, we assume $K_{i,j} = K_{j,i}$.)

When S_1 wants to initiate the transmission of a message m, it selects the first two nodes of a valid communication path, S_2 and S_3. It is required that S_2 is a 1-hop neighbour and S_3 is a 2-hop neighbour of S_1. (This is compatible with routing schemes that select shortest paths between communicating nodes.) S_1 creates two message authentication codes (MAC) for m using the shared keys $K_{1,2}$ and $K_{1,3}$: $a_{1,k} = \mathrm{MAC}(K_{1,k}, m)$ for $k \in \{2, 3\}$.

S_1 makes sure that the origin of m is included in the body of m. That is, the identity of S_1 is accessible through the component $m.o$ of m. Note that this identifier is part of the original message and cannot be changed later.

S_1 then transmits the data packet $d = \langle m \| a_{1,2} \| S_3 \| a_{1,3} \rangle$ to S_2.

When a node receives a data packet, there are two cases to consider. The first case is that the data packet was just initiated by the original sender. The other case is the more general case where an inner node on the path has to forward the message towards its destination.

The next step in our example is that S_2 receives the data packet $d = \langle m \| a_{1,2} \| S_3 \| a_{1,3} \rangle$ from S_1. S_2 checks whether $m.o = S_1$, i.e. if S_1 is the original source of the message. If this is the case, S_2 verifies that $a_{1,2} = \mathrm{MAC}(K_{1,2}, m)$. The latter step is a protection against the injection of forged messages on the link between S_1 and S_2. If successful, S_2 *accepts* the message m.

If S_2 accepts m and decides to forward it, S_2 constructs new MACs $a_{2,k} = \mathrm{MAC}(K_{2,k}, m)$ for $k \in \{3, 4\}$. It then sends a new data packet d' to S_3:

$$d' = \langle m \| S_1 \| a_{1,3} \| a_{2,3} \| S_4 \| a_{2,4} \rangle \ .$$

The following steps are performed by all consecutive, forwarding nodes. Assume that S_l receives a data packet in the general format

$$d = \langle m \| S_p \| a_{p,l} \| a_{q,l} \| S_r \| a_{q,r} \rangle \ .$$

S_l *accepts* m if the following two conditions are fulfilled by the contents of the data packet:

- S_p is a 2-hop neighbour of S_l and $a_{p,l} = \mathrm{MAC}(K_{p,l}, m)$
- S_q is a 1-hop neighbour of S_l and $a_{q,l} = \mathrm{MAC}(K_{q,l}, m)$

If S_l accepts the message, it continues forwarding it in the same manner.

Note that part of the contents, $a_{q,r}$, is opaque to S_l. S_l cannot verify if this MAC is correct and is not required to do so. S_l will simply forward the MAC to the next node.

Note also that a node cannot decide on the next node on the path by itself. This decision has been made already by the previous node. Rather, it decides on the second-next node. We assume that this incurs only a slight overhead on the routing layer. (But this point needs further investigation.)

Forwarding stops when the message has reached its final destination. The final destination is not always determined by the original sender, especially in content-based routing schemes. Note that constructing a broadcasting scheme out of the basic scheme is possible.

3.4 An Extension

A possible extension to the presented scheme is the addition of extra pairwise shared keys. This can be used to strengthen the resilience of the network against compromised nodes, which will be discussed in the following section.

Assume that a node A learns about other, remote nodes in the network (a set of node identifiers could also be chosen randomly and given to the node before deployment). A could initiate a key agreement protocol with any one of them (let's call the node B) and establish a shared key. Assuming source routing (the source completely determines the path a message will travel; this could be useful for persistent communication patterns), A includes a $\mathrm{MAC}(K_{A,B}, m)$ for a message m that travels a path that includes B. B will later accept m only if a valid MAC is included in the data packet.

We observe that if A is the original source of a message m and B the intended final destination, we obtain end-to-end security between these nodes.

4 Analysis

4.1 Reliability

It has been shown by Dolev et al. [3] that in the face of an adversary controlling t wires between a sender and a receiver, there must be $(2t + 1)$ wires avaialable to guarantee correct message delivery. In [1] it has been shown that costly communication paths can be substituted by cheaper authentication paths, while the reliability of a transmission is maintained. According to these results, the 4-connected network shown in Fig. 2 can tolerate a maximum of one compromised node. However, due to the special structure of the authentication path, two compromised nodes can be tolerated if they are not adjacent. Generally, an attacker must control two nodes to disrupt one path.

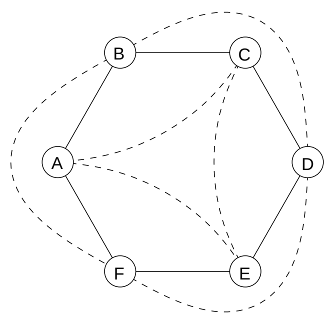

Figure 2: A simple sensor network with *Canvas*-style authentication paths

As an example, consider A sending a message to D. There are two paths the message travels: $\langle A, B, C, D\rangle$ and $\langle A, F, E, D\rangle$. If the attacker controls one of the intermediary nodes $\{B, C, E, F\}$, he cannot manipulate the message. If he controls two non-adjacent nodes, such as B and F, he still cannot manipulate the message. For carrying out a successful attack, he needs to control one of the pairs $\{(B, C), (E, F)\}$.

Due to this observation, we can expect that an attacker needs to compromise more nodes if the *Canvas* scheme is used than for the case where only linear authentication paths are available (as they were considered in [1]).

5 Simulations

We are testing the effectiveness of our approach by simulating random networks. The networks consist of 250 nodes, uniformly distributed on a rectangular plane (1000 m side-length), with a communication range of 100 m. We assume a routing scheme that uses exactly one (shortest) path between a pair of nodes. We say that a path is *lost* if it has a compromised node as an endpoint. (We disregard such paths, since any authentication scheme becomes ineffective on such paths.)

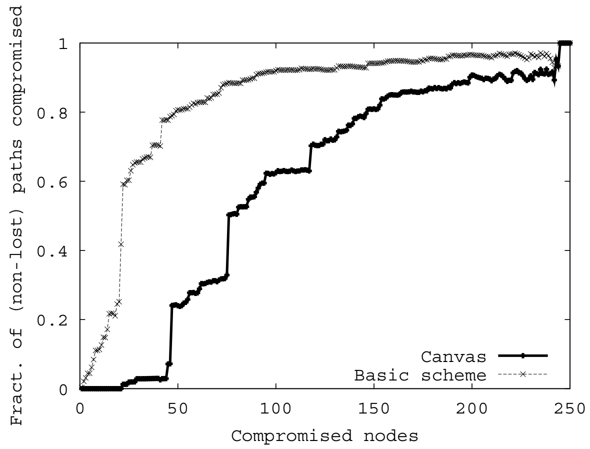

Figure 3: *Canvas* compared to a basic trust-chain approach

We are first comparing the *Canvas* scheme against a basic communication scheme, where only local links are authenticated and thus a path can be disrupted by a single malicious node (this is comparable to a trust chain). To assess the reliability of the network as a whole, we count the number of undisrupted paths, while an attack is going on. The attack is carried out by randomly picking single nodes and compromising them. In the basic scheme, this disruptes all paths that contain that node. Using the *Canvas* scheme, a path becomes inactive only if two adjacent nodes are compromised. As Fig. 3 shows, *Canvas* helps to extend the lifetime of a network under attack. With the basic scheme, 22 compromised nodes are sufficient to disrupt approximately half the communication paths. Using *Canvas*, these 22 nodes have almost no effect; instead, 76 compromised nodes are necessary to disrupt half the paths. Although these numbers depend heavily on the choice of the compromised nodes, they show the potential of *Canvas*.

6 Conclusion and Future Work

The *Canvas* scheme achieves data integrity at very low cost for sensor node communication. It relies on symmetric cryptographic operations and a low number of keys that have to be stored; it is therefore well-suited for resource-constrained sensor networks. We hope that it has become clear that in a large distributed system, such as a WSN, end-to-end security is not always necessary, and data integrity can be achieved with less effort.

The potential of *Canvas* is about to be explored more deeply, including:

- The use of multiple *Canvas* paths for the detection of attacks and for increasing the reliability of transmissions
- The transmission of confidential data over *Canvas* paths
- The exact cost of implementation for individual sensor nodes, including the power requirements compared to other approaches
- The relation and interaction of *Canvas* with routing protocols
- Applications of *Canvas* in other areas than sensor networks, such as ubiquitous computing environments

A more intelligent attacker than the random one considered in Sect. 5 will always try to attack the weakest areas of a network, e.g. to achieve a partitioning of the network. Therefore, we will also investigate how especially threatened nodes can be protected by adapting *Canvas*.

References

[1] A. Beimel and M. Franklin. Reliable Communication over Partially Authenticated Networks. *Theoretical Computer Science*, 220(1):185–210, 1999.

[2] J. S. L. Cheng and V. K. Wei. Defenses against the Truncation of Computation Results of Free-Roaming Agents. In *ICICS*, number 2513 in LNCS, pages 1–12. Springer-Verlag, 2002.

[3] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. *JACM*, 40(1), January 1993.

[4] L. Eschenauer and V. D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *CCS'02*. ACM, 2002.

[5] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. *Mobile Computing and Communications Review*, 1(2), 1997.

[6] Tim Kindberg and Kan Zhang. Context Authentication Using Constrained Channels. In *Fourth IEEE Workshop on Mobile Computing Systems and Applications*, pages 14–21. IEEE, 2002.

[7] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*, chapter 12. CRC Press, 1996.

[8] P. Papadimitratos and Z. J. Haas. Secure Data Transmission in Mobile Ad Hoc Networks. In *WiSe'03*. ACM, 2003.

[9] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Mobile Computing and Networking*. ACM, 2001.

[10] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *WiSe'03*. ACM, 2003.

[11] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing Pair-wise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In *Int. Conf. on Network Protocols (ICNP'03)*. IEEE, 2003.