How key establishment in medical sensor networks benefits from near field communication technology

Oliver Maye
IHP
Frankfurt (Oder), Germany
maye@ihp-microelectronics.com

Steffen Peter
IHP
Frankfurt (Oder), Germany
peter@ihp-microelectronics.com

Abstract— **A major design challenge for medical sensor networks (MSN) is to assure security of the communication links despite of the extremely scarce computational resources. From the security perspective, the establishment of trust during the initial key negotiation phase is a critical issue. The work presented is dedicated to the problem of initially pairing a network node with its sink. The proposed algorithm greatly benefits from near field communication (NFC) technology. Its small operating distance inherently introduces a close coupling between a node's physical presence and its logical certificate. Important advantage is drawn from the hierarchical architecture of typical MSNs.**

Keywords- medical sensor networks; key establishment; near field communication

I. INTRODUCTION

Medical sensor networks (MSN) are a sub-class of wireless sensor network (WSN) that is tailored to its primary application field in medical tele-monitoring. It is made up of so-called nodes – tiny, resource-limited minimalistic computing devices equipped with sensing capabilities and a wireless interface. The sensor enables a node to measure e.g. body temperature, blood pressure, oxygen saturation and alike. The form factor of a node is aimed to be coin-size and less.

There are certain nodes in an MSN with more powerful computational capabilities and possibly no sensors, called sinks. A sink collects measurement data from other nodes and aggregates it for further processing. We presume there is always one sink associated to each patient. Usually, this is unique in that MSN. However, e.g. in emergency situations it may be necessary to introduce more sinks into the MSN dynamically. The form factor of sink is like nowadays smart phones.

The intended use for monitoring a patient's vital data and other medical parameters implies an architecture that classifies MSNs as hierarchical WSNs. This means, that communication links between the entities of an MSN form a hierarchical graph. Each node directly communicates through a bidirectional link to one or more sinks. Each sink may or may not have an uplink to propagate aggregated data to a doctor's workstation. However, this uplink communication will be neglected for the rest of this work. In an MSN, there is no inter-node communication. The outline of a typical MSN is depicted in Figure 1.

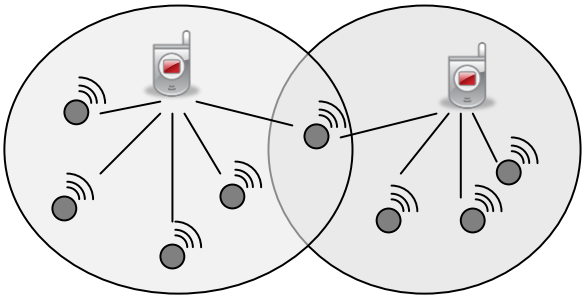

Figure 1. The general architecture of medical sensor networks. The cell phone icon stands for the data sink, whereas the small circles symbolize the nodes equipped with a wireless interface. Straight lines indicate physical communication links. One of the nodes is connected to two sinks, simultaneously.

Before an MSN can actually be used, several preparative steps must be passed. Together with the usage phase those steps comprise a kind of lifecycle. Regarding the lifecycle of an MSN, the following phases and terms are distinguished:

- Fabrication: This is the process of producing the nodes and sinks and all involved components. Fabrication completes with the manufacturer's delivery to e.g. the hospital.
- Initialization: Hospital-side preparation of new or re-used nodes and sinks for future use. May comprise of uploading firmware onto the nodes / application software onto the sinks, pre-configuring devices, recharging batteries etc. and is typically done by IT staff in a controlled environment.
- Pairing: Actually defines the participants in an MSN. Associates the nodes with their sink and vice versa and is typically performed by medical staff. Practically, the doctor decides on the nature, the number and placement of the sensors and – magically – makes them know each other and work together. This phase runs in a potentially unsafe environment.
- Use: In this stage, the MSN is established and delivers data to its supervisor. The usage period will typically last relatively long. It will roughly coincide with the corresponding patient's therapy. During usage, a node could be de-paired from a network or paired with an additional sink. As with the pairing, this phase runs in an uncontrolled, potentially unsafe environment. After usage, a node or sink will re-cycle through the initialization phase.

In this paper we present an approach that exploits beneficial properties of near field communication (NFC) to secure the pairing phase. That will be done with a trust-based pairing algorithm for establishing a secure link between a medical sensor node and its sink is proposed.

For this, the rest of this paper is organized as follows: Next, a definition of the original problem is given and the expected scientific contribution is sketched. Chapter IV surveys the state of the art and is followed by a description of the proposed solution in chapter V. This solution is analyzed and its pros and cons discussed in chapter VI. A summary and sneak preview to future work conclude the paper.

II. PROBLEM STATEMENT

The problem that this work is dedicated to is how the pairing can be achieved in a secure manner. Commonly accepted fundamental requirements of MSN are confidentiality, data integrity as well as access control and availability. So, more precisely, the problem is: How can the node and sink identify each other and set up a secure communication channel? Obviously, secure communication can be achieved efficiently through symmetric encryption. But still, exchanging the keys for that encryption trustworthy and with good performance is to be solved for resource-constrained devices.. Any satisfying solution must respect the very limited computational capabilities and power supplies, especially at the node-side.

III. AIMED CONTRIBUTION

The main contribution is to propose a practical and cost-efficient solution for the problem just stated and to discuss strengths and weaknesses of that solution. A technical realization and its benchmarking based on quantitative measurement results are subject to upcoming acts.

IV. RELATED WORK

Recently, Alemdar and Ersoy [1] surveyed the state of the art in the design of medical sensor networks. They identify the key exchange issue as a "challenge and open research problem" and reference an ECC based scheme proposed by Mišic in 2008 [2]. That involves a central trusted security server to generate all symmetric keys. Moreover, it lets the node authenticate itself via the primary wireless link, which does not enforce physical presence.

Dressler presents a concise overview of recent findings in key management for wireless sensor networks [3]. Approaches to key distribution are grouped into three classes: a) key pre-distribution, b) pro-active key distribution and c) on-demand key exchange. Classes a) and b) are introduced in more detail with various sample solutions analyzed and compared. Class c) covers public key solutions and applies to what will be proposed below, but, unfortunately, is not elaborated there any further. Obviously, at the time of publishing, work in this area was still ongoing.

Instead, Dressler references the group of Zitterbart et al. which began working on public key cryptography on resource constrained device quite early. They showed that it's feasible to deploy ECC-based public key cryptography on an Atmel ATmega128 – which is the base of the popular MICA2 platform – in 2005, already [4]

Huebner, Baldus and Garcia filed a patent application [5] on WSN key distribution. The core idea is an abstract key material box to perform key distribution to the involved nodes. The general procedure – node identification, key generation, encryption and transmission – tends to be similar. However, the node is identified by a handshake procedure based on a shared secret, i.e. symmetric key. Moreover, the concept is very generic and does not propose certain cryptographic algorithms for realization.

At the same time Garcia and Baldus proposed a lightweight security system involving a polynomial-based α-secure key establishment scheme [6]. Again, authentication is defined by the possession of correlated keying material. On the other hand, this approach allows a very efficient implementation. Their resulting prototype arrives at 25ms / 400 Byte RAM for a security handshake and will be a benchmark to compare with.

Camtepe [7, 8] studied key distribution in hierarchical WSN earlier and outlined the basic principle of pair-wise key distribution schemes. Here it is assumed that each node shares a pre-distributed secret with the base station, already, while the focus is to establish a secure communication link between any-two nodes. The general concept is what she calls "straightforward" and multiple approaches exist since the early 2000s.

McCusker [9] discussed the use of ECC for the asymmetric cryptographic part and proposed to implement the corresponding algorithms in hardware for the technical realization. Further elaboration of the protocol is not tailored to a specific application and, hence, misses the advantage that can be drawn from the characteristic architecture of MSNs. However, the resulting ECC hardware performs remarkably well.

Finally, Ng [10] has mentioned a trusted server scheme. Additionally, he argued that key pre-distribution schemes generally face a scalability problem subject to the network size. Findings are mentioned, that demonstrate the merits of ECC in key establishment for sensor nodes. More than this, the need for hardware implementation of ECC is foreseen.

V. AUTHENTICATION USING NFC

There are two types of entities involved in the pairing phase of an MSN: Nodes and sinks. As mentioned earlier, there is no inter-node communication. So each node needs exactly one key to secure the communication with its corresponding sink. This key is generated by the sink and given to the node during an initial pairing phase. To protect from unauthorized network participation, the node must authenticate with the sink, beforehand.

The core idea is to use near field communication (NFC) [11] technology for identification during the pairing phase. For the protocol proposed in the following we assume the node be equipped with a passive NFC tag and the sink be capable of reading such tags. During initialization and in a controlled, safe

and trustworthy environment, the following data is written to the node's tag:

- The node's public key E_N
- The node's addressing information A_N according to the wireless interface
- Both information signed using a trustworthy certificate, e.g. the hospital's signature $S_H(E_N, A_N)$

Furthermore, we assume the following data were stored on the node, possibly to its non-volatile memory:

- The node's private key D_N
- A trustworthy certificate over e.g. the hospital's public key E_H

Finally, before entering the pairing phase and still in the trustworthy environment, the sink is provided with the following data:

- An individual public/private key pair E_S, D_S allowing the sink to generate a signature S_S
- A trustworthy certificate over e.g. the hospital's public key E_H

Then, after initialization and back to the harsh clinic environment, the pairing phase is started by bringing the node physically near to the sink. This enables the sink to read the content of the node's NFC tag. By this, the sink learns the node's public key E_N as well as some addressing information A_N along with the proof of trust $S_H(E_N, A_N)$. That proof is made up of a cryptographic signature of the afore-mentioned data E_N and A_N issued by a trust-worthy authority, such as the hospital or the local public authority.

On successful reading a node's tag data, the sink checks the validity of the presented public key by verifying the authority's signature using the locally stored copy E_H of the hospital's public key. It calculates

$$E_H\left(S_H\left(E_N, A_N\right)\right) = H\left(E_N, A_N\right)$$

and compares the result with the hash value H' of the plain data E_N and A_N as read from the tag.

Next, it generates a random key K and encrypts this key using the node's public key: $E_N(K)$. The resulting data is signed with the sink's certificate which in turn was issued by the hospital. Finally, it sends the following data to the node via its primary wireless link (not NFC):

$$\left\{E_N\left(K\right), S_S\left(E_N\left(K\right)\right), E_S, S_H\left(E_S\right)\right\}$$

The node now checks the signature of the received data relying on the locally pre-installed hospital's certificate (E_H). It computes

$$E_H\left(S_H\left(E_S\right)\right) = H\left(E_S\right)$$

and compares the result on the hash value basis. Once trusting the sink's public key E_S, it makes use of this trust to gain confidence on the authenticity of the encrypted data, received simultaneously. Now, the node calculates

$$E_S\left(S_S\left(E_N\left(K\right)\right)\right) = H\left(E_N\left(K\right)\right)$$

and checks this against the hash value of $E_N(K)$.

On success, the node finally decrypts the data $E_N(K)$ using its private key D_N and, by this, retrieves the shared secret key K for further communication with the sink:

$$D_N\left(E_N\left(K\right)\right) = K$$

The whole process is visualized in Figure 2.

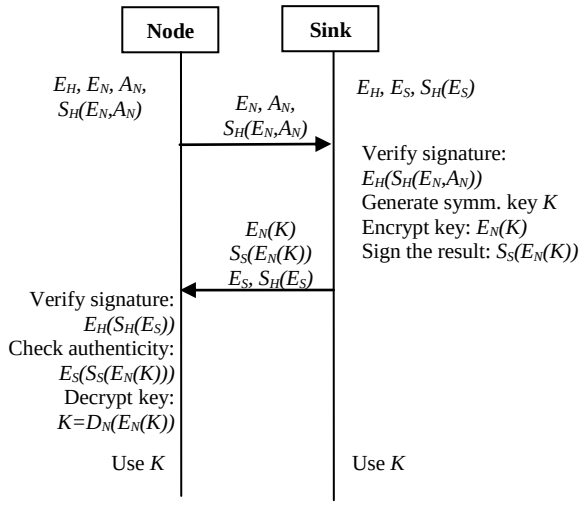

Figure 2. Message sequence chart of the setup process

In case the node shall be connected to another sink, the whole process is repeated with that sink. The result of the repetition is independent of what happened previously. So, the algorithm supports multiple connections of a single node to several sinks and vice versa. That enables a sink to span a true network of multiple nodes and allows an individual node be part of several networks.

VI. Analysis & Discussion

In what follows, the strengths and weaknesses of the procedure described above are discussed from the security perspective by running through the most probable threats.

The algorithm starts with its most vulnerable part – initializing sinks and nodes with their individual, private data. Most critical is the hospital's signature under the device's public key as well as the local copy of the hospital's public key. The trust established between the involved parties throughout the rest of the algorithm strongly depends on the trust-worthiness of the initialized data. This is where a hypothetical attacker named Eve brings leverage to bear on.

In an effort to forge or eavesdrop medical data, Eve could try to intrude the medical sensor network by introducing own, malicious nodes or sinks. Assuming Eve could succeed, the consequences would be fatal. A foreign sink could read

measurement data from the sensor nodes and thus, violate the patient's privacy. Even worse, it could misbehave in the overall medical process and report wrong data or nothing at all to the supervising system endangering the patient's life and wellbeing. The same risk is with foreign nodes, which would massively influence the therapy by reporting arbitrary data.

A. *Misuse the hospitals signature*

Eve creates an individual public/private key pair himself and puts it onto his node or sink device. Furthermore, he needs the hospital's signature under his public key to pretend certification. That would allow participating in the above mentioned protocol and get accepted by the communication partner.

In order to succeed, Eve needs access to the hospital's certification process, which mainly consists of applying the hospital's private key. So, here, access to the private key is crucial to introduce unauthorized devices and thus, is, what Eve must be prevented for. That's why, protecting the signing and certification process of the trustworthy authority (hospital) is very important. This is the reason, why the initialization phase has to be done in a secured, trustworthy environment excluding attackers like Eve, effectively. Practically, the hospital's certification process could be realized with dedicated computing hardware in a separate, physically safe IT chamber.

B. *Self-signed certificates*

Instead of misusing a valid signature, Eve could attempt to act as the certifying authority. As such, Eve could initialize own sink devices, as well as own nodes. Those devices would not (yet) participate in the medical sensor network as their certificate would be rejected during the authentication process. To overcome this, Eve would read a node's public key from its NFC tag, sign this public key and bring this certificate back onto the node's NFC tag, overwriting the original one. Then, Eve's sink could communicate with that node and read off the corresponding true data.

Despite of in this scenario, the sink of Eve will accept the node, it remains unsolved how the node can be convinced to accept the unknown certificate presented by Eve's sink. Even, if that could be accomplished, somehow, writing data to the NFC tag is Eve's next challenge. For that, Eve's writing device must be physically near the sensor, that is, in the vicinity of the patient or medical staff. Practically, this implies a high probability of being discovered. Further countermeasures are to write-protect the NFC tag with a passphrase or use read-only tags. Finally, a modified node falls out the original MSN and ceases delivery of data. That in turn would attract the attention of the monitoring software of staff.

If possible at all, this scenario can be detected rather quickly.

C. *Further security related observations*

After the initialization phase completed successfully, the remaining algorithm is as strong as its cryptographic components. The trust establishment algorithm described above relies on public key cryptography, a message digest (hash) algorithm and a symmetric key cipher. Therefore it is required that it is technically infeasible to:

- Calculate the private key from a given public key
- Calculate a hash collision
- Break the symmetric cipher

The technical realization must assure that this assumption is "practically true". That is, Eve cannot falsify the above assumptions with only a reasonable effort in time, money, computing resources etc. To achieve this, implementation should consider state of the art results of cryptographic research.

To further strengthen the security of the communication link between the node and its sink, both parties should replace the symmetric key periodically.

The algorithm described depends on a central trusted authority. Once the hospital – or other public authority – decides to replace their certificate, all devices must be re-initialized with that signature. That, in general, is a known drawback of PKI. To migrate more smoothly, the electronic stock of a hospital would be re-initialized MSN-wise, in practice. That takes advantage of a particular MSN's configuration being more or less static throughout a medical case.

Finally, the pairing algorithm described above does not guarantee authorization of the parties involved. Whether or not a certain sink is allowed to connect to a given node must be clarified in subsequent protocol steps. So far, the algorithm assures that both, node and sink, are controlled by the same trusted authority – the hospital. The focus here is to establish a secured link between a node and its sink.

D. *Technical realization*

As mentioned above, the implementation of the algorithm must assure that the procedure is practically secure. By nature, sensor nodes provide very limited computational resources. Nevertheless, in order to deploy the strongest possible cryptographic components known today, the use of elliptic curve cryptography (ECC) along with the secure hash algorithm (SHA) and advanced encryption standard (AES) is strongly encouraged.

All of them can be implemented efficiently in both, soft- and hardware. Moreover, a hardware implementation is regarded more power-efficient. When integrated with a low power multi-purpose processor this is considered the ideal base platform for sensor nodes targeting medical applications.

E. *Why NFC ?*

In fact, from the functional point of view, the role played by NFC in the above mentioned scheme could have been assigned to nearly an arbitrary radio technology. However, the intentional choice of NFC is motivated by the following reasons.

NFC naturally enforces spatial proximity. The fact that the node is near the sink when first presenting its certificate is an important asset in the way, how trust is established throughout the algorithm above. Surely, a very short communication range

can generally be achieved by lowering the transmit power of any radio technology. But the receiver side is unable to distinguish between a low-powered nearby signal from a normally-powered signal radiated from "far away". So it still could not be sure about physical vicinity. Introducing this possibility of a spatial gap potentially endangers the whole algorithm.

NFC tags don't need to be powered. The passive side of NFC does not stress the power consumption bill of the enclosing device, which, in the scheme above, is the node. This feature makes the node resist a denial-of-service attack consisting of continuously polling the node's certificate with the aim of draining the node's battery. In contrast to NFC, other radio technologies requiring an active, i.e. powered transmitter are vulnerable at this point, all the more, when transmitting data is much more expensive subject to power consumption, than receiving data.

NFC is an alternative medium just for key distribution. As a companion to the node's primary wireless link, NFC is "the other" medium used during initial key exchange, only. It's commonly accepted that using different channels for key exchange and data transmission improves robustness of the system from the security perspective.

NFC is cost-efficient. The passive tag for the node side is in the Euro-Cent range per unit. Even for the active part at the sink-side there are ready-built modules commercially available. First cell-phones with built-in NFC have entered the market a few years ago. The extra costs for NFC capability are comparable to those for other, well-established radio technology.

VII. Conclusion & Future Work

The authors proposed a novel, trust-based pairing algorithm for establishing a secure link between a medical sensor node and its sink. The approach applies NFC technology to establish trust in a practical and efficient way. As a result, the algorithm makes the sink be convinced of the certificate presented by the node truely belongs to that node, because of the physical vicinity necessary for reading a tag. The sink's influence on finishing the procedure successfully is slightly over-weighted. That corresponds to the computational effort and – well balanced – to the computational capabilities of the sink device.

Furthermore, attacks to the proposed procedure are discussed. From the security perspective, the initialization phase is most critical. That is, why it is important to generate and inject the crucial data, like certificates, to the node- and sink devices in a physically safe and trustworthy environment.

This work is in-progress. In the next step, the presented algorithm and ideas will be validated by a technical realization. For this phase, existing implementations of ECC, SHA-1 and AES in software and hardware will be used as cryptographic building blocks to accelerate the implementation. Based on this setup we can compare the eventual performance figures of the proposed algorithm to other, state-of-the art key exchange algorithms. The setup will also allow to evaluate the awareness of the approach in practice.

Future work will concentrate on integrating a hardware implementation of those algorithms with a low-power IPMS430 micro-controller. The outcome will be used as a hardware platform upon which to build wireless sensor nodes. The resulting node will be evaluated subject to its runtime-performance, power consumption, robustness and life-time period.

Acknowledgment

This work was funded by the German ministry for education and research (BMBF) under grant No. 01BS0801 and supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 225186.

References

[1] Hande Alemdar, Cem Ersoy, Wireless sensor networks for healthcare: A survey, Computer Networks, In Press, Corrected Proof, ISSN 1389-1286

[2] J. Mišic, Enforcing Patient Privacy in Healthcare WSNs Using ECC Implemented on 802.15.4 Beacon Enabled Clusters. In Proceedings of the 2008 Sixth Annual IEEE international Conference on Pervasive Computing and Communications (March 17 - 21, 2008). PERCOM. IEEE Computer Society, Washington, DC, pp. 686-691.

[3] Falko Dressler, Key Management in Wireless Sensor Networks, in Security in Wireless Mesh Networks, Wireless Networks and Mobile Communications, vol. 6, Yan Zhang, Jun Zheng and Honglin Hu (Eds.), CRC Press, 2008, pp. 491-516

[4] Erik-Oliver Blaß, Martina Zitterbart: Towards Acceptable Public-Key Encryption in Sensor Networks, ACM 2nd International Workshop on Ubiquitous Computing, p. 88--93, INSTICC Press, Miami, USA, May 2005. (ISBN 972-8865-24-4)

[5] Axel G. Huebner, Heribert Baldus, Oscar Garcia, Wireless Sensor Network Key Distribution, PCT International patent application no. PCT/IB2008/051168, International Publication Number WO 2008/122906, 16 October 2008

[6] O. Garcia-Morchon and H. Baldus, Efficient distributed security for wireless medical sensor networks, in 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing. IEEE, December 2008, pp. 249–254.

[7] Seyit A. Çamtepe, Bülent Yener, Key Distribution Mechanisms for Wireless Sensor Networks: a Survey, Technical Report TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[8] Seyit A. Camtepe, Design and Analysis of Key Management Schemes for Distributed Wireless Sensor Networks, Ph.D. Thesis, Rensselaer Polytechnic Institute, Computer Science Department, May 2007.

[9] Kealan McCusker, Cryptographic key distribution in wireless sensor networks: a hardware perspective. PhD thesis, Dublin City University, 2008

[10] H S Ng, M L Sim, C M Tan, Security issues of wireless sensor networks in healthcare applications, BT Technology Journal, Vol. 24 No. 2, Apr. 2006, pp.138-144, ISSN 1358-3948, Springer Netherlands

[11] Near Field Communication, Interface and Protocol (NFCIP-1), ISO 18092:2004