Journal of Information Assurance and Security 2 (2007) 189–199

Secure and Trusted in-network Data Processing in Wireless Sensor Networks: a Survey

Alessandro Sorniotti, Laurent Gomez, Konrad Wrona and Lorenzo Odorico
SAP Labs France SAS
805, avenue du Dr. Maurice Donat
06250 Mougins, France
Email:name.surname@sap.com

Abstract: In-network data processing in wireless sensor networks (WSN) is a rapidly emerging research topic. The distributed processing could have several advantages for wireless sensor networks. First of all, in WSN computation is typically much less energy consuming than communication. Secondly, in-network processing enables WSN to provide more complex services to application layer, and not only data gathering functionality. However, in addition to computational overhead, in-network data processing introduces also many challenging security issues. Most of them are still open and require development of innovative security mechanisms. In this article we survey the current research related to security of in-network data processing in wireless sensor networks and highlight the directions, which are most promising in our opinion.

1 Introduction

Wireless sensor networks (WSN) introduce several technical challenges (16, 59). Some of the most important issues are related to extracting useful, reliable and timely information from the deployed sensor network, and include distributed information processing, data fusion and security (40). In-network data processing is defined as a processing the network nodes carry out on data in a distributed manner. The process is performed while data is exchanged, i.e. before it is acquired and used by the higher layers. Such distributed processing could have several advantages for wireless sensor networks. First of all, in WSN, computation is typically much less energy consuming than communication. In fact, on a power-constrained sensor, sending one bit requires almost the same amount of energy as executing 50 to 150 instructions (52). Hence, one of the goals is of course to reduce the traffic as much as possible, e.g. by exploiting the high redundancy and correlation that is normally present in sensor data. Secondly, in-network processing enables sensor networks to provide services, and not just data, and to fully exploit its multitude and redundancy properties. Therefore in-network data processing, such as data fusion and aggregation (53), has emerged in the recent years as an active research area in WSNs. One of the important issues related to in-network data processing is to find a realistic balance between computational overhead, delay, data resolution, and data trustworthiness. Therefore, in-network data processing in wireless sensor networks requires development of new secure and energy-efficient methods enabling fusion of relative large amount of data.

In our survey we focus on security and trust aspects of in-network data processing. In Section 2 we present rationales behind in-network data processing in WSN. In Section 3 we describe the main trust challenges in WSN. In Section 4 we introduce various security mechanisms, which can be used to address these challenges. In Sections 5 and 6 we discuss the in-network data processing and data aggregation. Finally, in Section 7 we briefly reiterate our main conclusions.

2 Secure Data Processing in Wireless Sensor Networks

Sensor networks are often described as the next hype technology of the 21st century (10, 56). Its ability to monitor and control diverse physical environments, ranging from battlefield to human body, makes them attractive for multitude of application domains, including military, health care and traffic control (16).

Most of the ongoing research activities in WSN focus on the lower layers, namely radio communication, routing and self-organization. But first on top of these layers the real capabilities of WSN are unleashed. Indeed, WSN can be seen not just as data source, but rather as a provider of services tightly related to data collection and processing. It is clear that WSNs must collect data, but the real added value lays in the fact that the multitude of nodes can also process this data. This is what is called in-network data processing (19): this processing can range from concatenation, aggregation, and simple mathematical operations such as averaging to more complex reasoning on data. The in-network processing layer represents one of the real strengths of WSNs: it can deliver not just raw data but also process it in real-time with no intervention of the infrastructure and leverage on redundancy to cope with data loss and corruption. In-network data processing can also be used to reduce the amount of information transmitted, as data coming from multiple sensors is usually highly correlated, and thus to achieve a longer network lifetime.

However appealing and powerful, in-network processing requires a high level of security: tampering with data at this level can indeed introduce threats that range from simple unauthorized data access to malicious data modification. The first line of defense against these threats are cryptographic mechanisms: integrity and confidentiality can be achieved using cryptographic schemes. The field of cryptography within in-network data processing (what we call secure data processing) is a very promising research field, and introduces many interesting challenges. For instance, a straightforward

Received September 12, 2007

approach is to achieve hop-by-hop encryption and integrity protection. However, nodes can be captured and the disclosure of their key material can lead to the disclosure of raw data and partial results, or malicious modification of data. In particular, the packaging of sensor nodes can also be affected by the low-cost requirements, not allowing for tamper-resistant devices: hence, nodes may be captured and their secret material can be disclosed to an attacker. To overcome this problem, techniques are proposed, that exploit end-to-end encryption in conjunction with particular key distribution mechanisms, homomorphic encryption schemes or public cryptographic schemes. Nevertheless, in order to make sensor networks economically viable, sensor devices are limited in their energy, computation and communication capabilities; hence these schemes have to take into account the technical and economic constraints.

Once data has been sensed and possibly processed by nodes, it must be delivered to data sinks, i.e. nodes that are responsible for gathering data and passing it to application gateways. Application gateways are nodes that are responsible to deliver data to the real point of exploitation. The sink nodes and gateways may suffer constitute single point of failure, given that they are normally much less numerous than wireless nodes. Sinks and gateways, being the eventual destination of data, are also the perfect point of attack in order to get access to data, modify it or supply false data. Countermeasures to such attacks include mutual authentication of the gateway and the nodes, ensuring that both the gateway is entitled to receive data and that the data is sent by legitimate nodes.

Higher layers of a WSN deployment include a middleware layer and an application layer. In addition to providing important functional service, WSN can be seen also as a source of external risks to the middleware and applications, as network layer can be used as means to attack them. A straightforward example can be the usage of a captured sensor node to supply a particularly crafted input to the middleware layer, in order to exploit software vulnerabilities such as buffer overflows.

Cross-layer security considerations in WSN are often related to the limited resources of sensor nodes in terms of energy, computation and communication capabilities. We want to stress however, that this might not always be true: indeed WSNs can also be deployed in scenarios that justify more powerful nodes, e.g. in automotive. If limited resources is the case, then every layer can be exposed to over-consumption attacks. Such denial of service attacks do not necessarily just focus on depletion of batteries, but also of memory or computational resources. The depletion attacks normally aim at requesting a huge amount of non-legitimate work by the sensor nodes in so that there are no resources left for legitimate requests.

Other risks can be categorized as general risks associated with the usage of WSNs. One of them is the so-called function creep. A function creep is what occurs when an item, process, or procedure designed for a specific purpose ends up serving another purpose for which it was never planned to perform. For instance, an ubiquitously deployed WSNs can allow secret surveillance, which was not planned as one of the intended usages. The protection against such threat is not straightforward since surveillance can take many different forms. Moreover, the countermeasures can require an implementation and enforcement of legal mechanisms and not just technical solutions.

3 Trust Challenges in Wireless Sensor Network

Sensor nodes used in typical WSN are not tamper-resistant devices (63), mostly due to the cost and power constraints. Weak physical security protection implies that an attacker can relatively easy capture and analyze nodes or introduce new malicious nodes. When a node is compromised, all the cryptographic material is disclosed to the attacker. This material is typically used for encryption and authentication of exchanged sensor data. By capturing the cryptographic material, the attacker can generate new malicious code including proper cryptographic mechanisms. The attacker can also disseminate forged sensor data in a properly encrypted and authenticated manner. His objective is often to disrupt the normal behavior of the WSN and compromise in-network data processing. Depending on his strategy, the attacker can influence in-network data processing in long-term or in short term. In the case of long-term attacks, the attacker can aim at skewing in-network data processing without being detected. It permits him to control the WSN behavior, and to influence application decisions based on in-network data processing. In the case of short term attacks, the attacker might not care about being detected, and aim at making the WSN inoperable.

An alternative approach to the problem of compromised nodes consists of evaluating trustworthiness of sensor data. When collecting or aggregating sensor data, we aim at computing the distance between the real and the delivered sensor value. This distance is related to sensor characteristics (e.g. accuracy, quality of service, resilience to failure), its reputation in the WSN, and the sensor data value itself. Few approaches have been already proposed in the literature for determining the trustworthiness of in-network processing of sensor data. Those approaches will be discussed in Section 6. In-network evaluation of trustworthiness of sensor data is based, e.g., on probability theory. The trust information can thus be used in order to determine if the sensor data should be skipped or be used for further data processing. Unfortunately, the scalability of those approaches is still questionable. Actually, the efficiency of those approaches is tightly related to the number of nodes involved in the WSN.

4 Security and Trust Primitives

In this section we will introduce the basic security and trust primitives that are needed to achieve secure and trusted in-network data processing. Some of the common cryptographic schemes are not suitable for WSN environment due to the particular characteristics of WSNs. For instance, limited energy supply calls for usage of the shortest possible keys required in order to achieve the adequate level of security. The lack of tamper-resistance calls for mechanisms where the capture of intermediate nodes does not allow to disclose all sensor data, or for mechanisms for determining the trustworthiness of sensor data.

The first primitive that we introduce is Elliptic Curve Cryptography (ECC). Elliptic curve cryptography is an approach to public key cryptography based on the algebraic structure of

elliptic curves over finite fields. The ECC was first introduced in (34) and (44). As we will explain in Section 4.1, the ECC is suitable for WSNs because of the reduced key-size when compared to classic public-key cryptography system with the same security level.

In addition, we will also introduce bilinear pairings (20), (21) and (23). Bilinear pairings are admissible maps of the group of points of a suitable elliptic curve; they possess particular properties such that they can implement oracles for the Decisional Diffie-Hellman (DDH) problem, yet keeping the Computational Diffie-Hellman (CDH) problem a hard one. This property can be useful in many circumstances, as we will see in Section 5.

Another family of primitives that we will introduce is Privacy Homomorphisms (PH). An homomorphism is a structure-preserving map between two algebraic structures: if the map is also an encryption scheme, then we call it Privacy Homomorphism. Generally speaking, PH is a tool that allow us to perform calculations on encrypted data. Privacy homomorphisms (PH) were first introduced by Rivest et al. in (54).

Regarding trust primitives for in-network data processing, we introduce an existing metric for trust evaluation based on beta distribution, which encompasses the notion of uncertainty. This metric is not particularly related to sensor data, but it supports determination of the probability that the processed data matches the real data. We also present subjective logic, which is based on a notion of *opinion* derived from the evidence theory (58) and supports operators for manipulating opinions about sensor data trustworthiness during in-network data processing.

4.1 Elliptic curves and bilinear pairings

The integration of ECC in wireless sensor nodes has been a research topic of central importance in last years. The main question in recent years has always been the same: is ECC suitable for WSNs? The first studies in this area started with the works documented in (11) and (29), conducted in 2000 and 2002 respectively. Unfortunately, the heavy energy requirements of public-key algorithms, reported in the above works, have raised serious concerns about the feasibility of ECC deployment in WSNs. However, since then, an enormous progress has been made in the efficient implementation of ECC (27). This progress allows to completely re-evaluate the energy requirements of elliptic curve cryptosystems and their applicability to WSNs. In fact, the energy cost of ECC is far lower than earlier believed, and state-of-the-art algorithms now allow researchers to conduct further experiments on a possible integration of ECC over WSNs.

Typically, nodes in WSNs cannot afford to run conventional public key cryptography (PKC). By using ECC it has been shown that PKC is feasible in WSNs since ECC, for a given security level, consumes considerably less resources than conventional PKC. This is in particular due to use of much shorter keys when compared to, e.g., RSA. Lenstra and Verheul (39), in fact, showed that 163 bits long key sizes for ECC correspond to RSA keys that are much longer than 1024 bits. More precisely, one could achieve that level of security with around 130 bits long ECC keys. This suggests that ECC seems to be very suitable for ubiquitous computing devices, requiring exchange of less data in the system, when compared to other cryptographic methods for the same level of security.

Many cryptographic applications based on elliptic curves use bilinear pairings, a powerful mathematical tool which enables efficient implementation of ID-based cryptography. Bilinear pairings allow Diffie-Hellman problem, a well-known class of hard computational problems, to have an easy decisional version. The question whether a quadruple (a, b, c, d) is a Diffie-Hellman instance, which is a hard decisional problem in the general case, can be efficiently answered in case of bilinear pairings. The difference between the decisional version and the computational version of a cryptographically useful NP problem enables radically new ways to implement security algorithms. But ECC implemented with bilinear pairings introduces an overhead, which has to be taken into account: the particularly time consuming part of the scheme is a pairing computation.

Despite of all challenges, WSNs introduce suitable scenarios for implementing security algorithms based on ECC. For example, pairings-based cryptography schemes such as Identity-Based Encryption (IBE) have strong requirements such as the existence of an unconditionally trusted entity, that is responsible for issuing users private keys. Fortunately, most of proposed WSNs possess such an entity, i.e. the sink node. Another requirement of IBE is that the keys must be delivered over confidential and authentic channels to users. In most of the WSN applications such private keys can be distributed offline, i.e. they can be generated and preloaded directly into nodes prior to deployment.

In spite of all its advantages, ECC-based schemes such as IBE still are a public key cryptosystem and thus they are orders of magnitude more complex than symmetric cryptosystems. Even though results of TinySA implementation of pairings for resource constrained nodes (24) show that strong elliptic curve cryptography is feasible on sensor nodes, its energy requirements are still much higher than that of symmetric cryptosystems. The authors of (24) explicitly underline that TinySA must use elliptic curve cryptography only for infrequent, but security critical, operations such as key establishment during the initial configuration of the sensor network or the authentication of routing information. Nevertheless, experiments with TinySA and others cryptosystems show that ECC, when implemented properly, is a valuable tool to improve the security of wireless sensor networks.

These first experiments open new possibilities for implementation of some security services with ECC, such as key-distribution, authentication, data encryption and access control. As we will see in the next section, ECC can also be used for its homomorphic properties.

In the recent years several experiments were focused on implementations of security policies based on ECC. These experiments were usually conducted with 8-bit or 16-bit CPUs. It was shown that ECC based PKC is feasible on sensor nodes or RFID tags with a hardware implementation of the security protocols. In (62), Wang et al. implemented an elliptic curve cryptography-based access control in sensor networks, with a 160-bit ECC implementation on Atmel ATmega128, a CPU of 8MHz and 8 bits. They showed that an ECC point multiplication takes less than one second with that hardware. With the same CPU and only 4KByte RAM, Carman et al. (11) obtained 6.88s for ECDSA signature and 24.17s for ECDSA verification. Other results described in (47) were measured

on a MICAz node running TinyOS. The obtained average execution time to compute a pairing was 30.21s. The costs concerning RAM and ROM flash memory were 1 831 and 18 384 bytes, respectively. Gura et al. in (26) documented their experiments with an Atmel ATmega128 at 8 MHz. They made a comparison between ECC and RSA on this 8-bit CPU in order to evaluate the execution time and the memory cost for different operations with these two approaches. They verified that a 160-bit ECC point multiplication requires around half of the memory space and a similar execution time of a RSA-1024 operation. The relative performance advantage of ECC point multiplication over RSA modular exponentiation increases with the decrease in the processor word size and the increase in the key size.

The two main parameters that are used to characterize an ECC system are the key size and the security multiplier. Concerning the key size, for most PBC schemes (including IBE) security requirements can be satisfied by choosing it equal to 160 bits. However, in WSN the system security requirements are often relaxed (51) in order to increase efficiency. This is often possible because of the relatively short system lifetime, especially when the goal is not to protect each node individually, but the network operation as a whole. The largest broken Elliptic Curve Discrete Logarithmic Problem yet had 109 bits key size over the finite field $F(2^{109})$ and it took 17 months (13) to break it. Instead, the largest broken Discrete Logarithmic Problem yet had 160 digits key size (33). Therefore, it seems that even 128 bits ECC key size is able to secure sensitive information in sensor networks.

For example, in (3) a curve over $F(2^{113})$ was chosen as it offers about 16 times more security than 109 bits, which seems enough security for todays hardware. Batina et al. (2) assumed in their work that ECC over $F(2^{131})$ provides a good level of security for their application.

The key size is not the only important parameter. In fact, during the creation of the system one can choose a specific value of K, also called the security multiplier K. The value of K is crucial as all the pairings computations are actually going to be performed in $E(F_{p^k})$, with E being an elliptic curve studied over F_{p^k}.

A pairing can be computed efficiently if K is small. Supersingular curves are a particular kind of curves, which impose a small number of possible group structures and depend on the number of points in the $E(F_{p^k})$. Supersingular curves, although in the past used to be avoided in cryptography because they are more vulnerable to some specific attacks, let the security multiplier be $K \leq 6$. It is known from (60) that, with $K = 2$, 1024 bits finite field is roughly equivalent to 512 bits on the corresponding elliptic curve. If $K = 3$, it is equivalent to 340 bits. If K becomes big, the ratio is not the same: 1024 bits security on finite field is not equivalent to 51 bit security on elliptic curves, it is rather equivalent to 160 bits.

For non-supersingular curves, K could be greater ($10 \leq K \leq 50$). If the security parameter is large it is good from a security point of view, because it is directly proportional to the system level of security, but the computation will be hard. The choice of K thus depends on these two properties.

4.2 Privacy homomorphism

A privacy homomorphism is a family of functions $(e_k, d_k, \alpha, \gamma)$, such that

$$d_k(\gamma(e_k(m1), \ldots, e_k(mr))) = \alpha(m1, \ldots, mr)$$

for each key k in some key space and for any m in some message space.

Privacy homomorphisms (PH) were first introduced in (54). The authors already noticed that if a given PH allows to evaluate the $\leq$ predicate and allows an attacker to generate encrypted versions of constants, then the scheme does not provide confidentiality, as a binary search strategy can easily reveal the encrypted value. Although the schemes proposed in (54) were broken in (7), PHs have been since then an important research topic.

The two most common variations of PHs are the additive PH and the multiplicative PH. The latter provides the property

$$e_k(m1 \times m2) = e_k(m1) \oplus e_k(m2).$$

Well known examples of multiplicative PHs are RSA and the discrete logarithm ElGamal. Additive PHs provide the property

$$e_k(m1 + m2) = e_k(m1) \oplus e_k(m2)$$

and are useful since they can be used to calculate, e.g., the average value.

A first very simple family of privacy homomorphisms are simple variations of the one-time pad scheme. An example of such scheme is

$$e_k(x) = (x + k) \, mod \, n$$

The security of such scheme relies on the one time use of the key and n is a publicly known value. Although this scheme is provably secure, it should be complemented with mechanisms to create a secure key stream that must be used only once. This family of PH schemes allows the computation of the sum of encrypted values.

A large subgroup of PH cryptographic algorithms is based on high degree residuosity (50). These schemes provide the additive PH, but need very long keys that in turn imply large messages and computation effort. In (5) the properties of finite bilinear groups with composite order are used in order to construct a new scheme that allows to compute a single multiplication on the encrypted data, along with an arbitrary number of sums.

A symmetric PH scheme with both the additive and the multiplicative PH property, which makes it an algebraic PH, was introduced in (18). It is a symmetric algorithm that requires the same secret key for encryption and decryption. The scheme generates a vector of d integers that sum up to the cleartext value; these integers are then multiplied by the secret key, risen to all the powers in the interval $[1, d]$. The aggregation is performed with a key that can be publicly known. The same secret key must be distributed to every node in the network. The message size is proportional to the parameter d, so that for $d > 100$, the messages become very large. This scheme has been broken in (15), where it has been shown that it is possible to break the scheme given $d + 1$ known plaintexts: we underline that this assumption can be easily met in a WSN setting, were it is possible to deploy a fake sensor that measures the same value of a rightful one and in the same time eavesdrop the encrypted message generated by the

latter. It is generally agreed that privacy homomorphisms are weak under chosen-plaintext attacks and some of them have been broken by weaker attacks such as known-plaintext or even ciphertext-only attacks.

In contrast to PHs discussed above, the PHs based on elliptic curve ElGamal (ECEG) rely on an asymmetric cryptography. The benefit of this PH is that the encryption key may be publicly known. The ECEG cryptographic algorithm requires that the message text must be mapped on the EC space. An approach has been proposed that multiplies the message text m with the generator of the EC G. A problem with such solution is that the decryption leads again to the mapped point $m \cdot G$, but it is not trivial to compute m out of $m \cdot G$. Since it is the fundamental property of ECC that the point multiplication is not efficiently invertible, the only solution is a brute force computation that relies on a limited domain of the mapping. In most cases this approach is very reasonable.

Another family of homomorphic cryptographic primitives are homomorphic signature schemes (31). The first type of such schemes are redactable signatures: given a text signed with a signature, we can produce the signature for a subset of the text (for instance, given a sentence and its signature, we can produce - without access to the private key - the signature for the given sentence with some words missing). Another type, set-homomorphic signatures, when given the signature over sets of elements allow to produce signatures over the union of these sets.

4.3 Subjective Logic

Subjective logic is based on Dempster-Schafer theory of evidence (58) and enables handling of opinions about propositions. An opinion is represented by a 4-tuple (b, d, u, a), where a represents the a priori probability in absence of opinion. As we only consider binary state space for proposition P, we assume $a = frac12$. Respectively, b, d and u represent the belief that P is true, the belief that P is false, and the uncertainty is the amount of belief that is no committed to the truth or falsehood of P. The range of those four value is $[0, 1]$ where

$$b + d + u = 1.$$

The opinion of A about P is defined as

$$\omega_P^A = b + au.$$

Moreover, subjective logic framework provides a set of logical operators for combining opinions, such as conjunction, disjunction and negation. Subjective logic supports also non-traditional operators such as average or discount of opinions (32).

Applying subjective logic to evaluation of trustworthiness of sensor data consists of determining opinion about the following proposition : *the sensor data is trustworthy enough to be used for intended application.* In addition, subjective logic permits to represent the uncertainty about quality and accuracy of sensor data. Thus, in-network data processing can benefit from subjective logic operators for combining opinions on collected sensor data. The use of beta probability density function in subjective logic enables also the establishment of opinions on sensor data based on sensor reputation (30). In beta probability density function $\phi(p|a, b)$, the positive and negative feedback on sensor node are determined by a and b. In the scope of sensor data trustworthiness, beta distribution provides sound mathematical tool for determining the future evolution of the sensor data. Following the example presented in (30), given 7 outcomes close to real-value and 1 outcome different to real-value, we have a beta function defined by $f(p|8, 2)$, plotted in Figure 1. We can calculate the probability $E(p) = 0.8$ that an outcome is close to real-value.

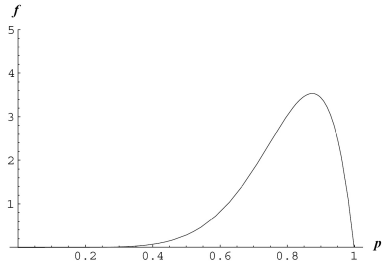

Figure 1: Beta Distribution for $f(p|8, 2)$ (30)

5 Secure in-network data processing

Although most of the research in the security of in-network data processing is quite recent, it has produced many promising results. As we already discussed, a topic that has attracted particular attention is homomorphism of cryptographic functions such as encryption or signatures. These PH techniques provide foundations for adding security to in-network data processing. The security goals of in-network data processing are mainly confidentiality, integrity and authentication of data origin. The operations involved in in-network data processing range from concatenation of data and mathematical operations (mainly addition and multiplication) to operations on sets (such as subset queries or comparison queries).

5.1 Concealed data aggregation techniques

Concealed data processing techniques aim at processing sensor data while protecting confidentiality of both raw data and intermediate results. In (57) two general approaches are presented: hop-by-hop encrypted data aggregation and end-to-end encrypted data aggregation. The former is easier to achieve, but it leaves aggregator nodes vulnerable to attacks because the sensor readings are decrypted by those aggregators. In the latter, the intermediate aggregation nodes do not have the decryption keys and can only perform some operation on encrypted data. In the hop-by-hop scenario, a bootstrapping phase is needed, which consists of either pair-wise or group-wise key distribution schemes. Once keys are shared between nodes, data is decrypted, processed, encrypted and sent on toward a sink. In the end-to-end scenario, a network-wise key needs to be established between the sink and all the sensor nodes. This can be achieved using a master key or a public-key based solution. The additional security with respect to the hop-by-hop scenario is that all the sensors have the network key, but aggregators do not. Once the key is established, one of the privacy homomorphisms introduced in Section 4.2 can be used to aggregate the data.

In (48), a technique for end-to-end secure data aggregation is described. The network is modeled as a tree rooted in a sink. The encryption algorithm is an additive privacy

homomorphism based on the standard one time pad. The pseudo-random key stream is generated by successive encryption, using a random key, of a counter. Nodes are organized in layers: a node belonging to a layer has $n - k$ keys, while there are n layers of encryption. When data is processed at a given layer, $n - k$ layers of encryption are stripped, i.e. a node performs the decryption process $n - k$ times using the keys it possesses. The encrypted value is processed - thanks to the privacy homomorphism - and $n - k$ layers are then restored. The advantage of this scheme is that if nodes are compromised, it is still impossible to disclose partial information because there are always k layers of encryption to protect the data. The drawback of this approach is that it heavily relies on the key pre distribution phase and a very rigid network topology.

In (12) another end-to-end technique for secure data aggregation is presented. This scheme is similar to the one presented before: the homomorphic function used to perform the computation is again a simple variation of a one-time pad scheme and the network model is a tree rooted at the sink. In the previous scheme, a group of nodes belonging to a given layer shared a set of keys; in turn, in this scheme, each node shares with the sink a single, distinct, long-term key; with this key, each node generates the keystream used to encrypt its data. Every node can then aggregate encrypted data thanks to the homomorphic properties of the encryption scheme. Even if a node is compromised and its secret material revealed, the secrecy of other nodes' data and of temporary results is preserved, since it is still protected by the encryption performed by other nodes.

In (41) the network is organized in clusters and the aggregation is performed only by cluster heads. Similarly to the previous two solutions, key distribution is required. The authors present a protocol for establishing cluster keys using ECC: each node has just a share of the secret key, whereas the public key is publicly known. An attacker that can compromise up to t sensors out of n, cannot recover the secret cluster key and therefore cannot access sensor data. Cluster head can safely decrypt and aggregate the measures. This scheme protects against compromised nodes, however it does not protect against compromised cluster heads. Nevertheless, in many cases it may be reasonable to assume that cluster heads are more powerful (possibly also tamper resistant) nodes, so it is harder for an attacker to compromise them.

5.2 Integrity of aggregation

When sensor data is processed, in addition to confidentiality an important goal is to ensure that the result of the process is actually the intended computation and not a maliciously crafted value. In (57), several techniques for certifying that the outcome of the aggregation is actually based on sensor data are surveyed: such techniques generally rely on shared keys between sinks and the nodes. Each node produces a message authentication code (MAC) based on a temporary key, which is derived from the shared key. MACs are aggregated, when data is processed, using a logical tree structure, rooted in the sink. The sink can verify the final result of aggregation and broadcast the used temporary keys, so that each node can verify the intermediate aggregation results. Variations of this scheme use Merkle hash trees as commitment structures for the aggregated values.

One of the security goals of the technique proposed in (41) is to ensure that cluster heads do not accept faulty readings for an upperbound of t compromised sensors. The verification of the latter uses threshold cryptography. The result of the aggregation (i.e. average value) is checked against nodes' own readings: each node checks whether the difference between the calculated average and its reading is within a given threshold, in which case it generates a partial signature. A valid signature can be generated by $t + 1$ nodes. In order to limit the number of hash values, which need to be computed and checked at the cluster head, a Merkle hash tree is built.

In (14), the authors propose a mechanism to check that in-network aggregation in WSN is complete, i.e. that no contribution from any node has been excluded from the final result. Completeness is enforced by creating a logical balanced tree structure on top of the sensor mesh. The tree should be balanced in order to reduce traffic overhead. Using this structure, all the contributions are authenticated with a hash function and propagated up toward the tree root. At the end of this phase, the root sends - via an authenticated broadcast (e.g. using TESLA) - the root of the hash tree; each node sends the value of the commitment tree of its child to all the other children and so forth, to make all the off-path nodes available to everybody, in order to verify the consistency of the hash tree. The commitment includes a lowerbound and an upperbound on the value of the sum, so that a node can verify that an attacker has not altered the valued. Once a node has checked that its contribution was included, it sends a MAC to the root with a key it shares with it. All the confirmations are XORed to save space and produce a single confirmation message.

5.3 Signature aggregation techniques

Signature aggregation techniques are the technical answer to the need of creating a single signature for a single message out of multiple signatures for multiple messages. In (4) two signature aggregation techniques are presented: (1) a general one, where at any point, any number of signatures, can be aggregated by any node into a single signature; and (2) a sequential one, where aggregation has to be performed sequentially by signers during the process of signing. The first solution is based on two well-known problems, the Computational Diffie-Hellman (CDH) and the Decisional Diffie-Hellman (DDH). For many choices of groups, both assumptions hold (i.e. both problems are hard to be solved), but for some, CDH is hard but DDH is not. The groups that have this property are called Gap Diffie-Hellman groups (GDH). Based on this, the algorithm (1) is introduced. The sequential one (2), instead, is based on trapdoor homomorphic permutations (such as RSA).

5.4 Operations on sets

Often it can be interesting to perform set operations on encrypted data: in particular set membership queries, comparison queries, subset queries and arbitrary conjunctive queries. This family of operations is important if WSN support event-based paradigms: events can be seen as elements and sets can be constructed according to events' semantics and associated to actions that must be taken accordingly.

In (45) the authors suggest a way of creating a secure representation of sets. In the proposed solution, an entity is able to check only whether or not it belongs to the set – it is

impossible to check if another entity does. The set representation is constructed solving a system of linear Diophantine equations using the Chinese Reminder Theorem.

In (6) the authors propose a mechanism based on Hidden Vector Encryption (HVE). By using HVE a public key system supporting queries on encrypted data can produce tokens for testing any supported query predicate. The token lets anyone test the predicate on a given ciphertext without learning any other information about the plaintext. The proposed solution allows for comparisons and subset queries as well as conjunctive versions of these predicates.

6 Trusted in-network data aggregation

As explained in Section 3, compromised nodes represent a big threat to the security of in-network data processing. The challenge arises from the fact that sensor nodes often need to be low-cost to justify their deployment, which makes it very hard to satisfy tamper-resistance requirements. An attacker could gain control over a sensor node in a stealthy way in order to generate faulty data or to alter the data processing. Thus, once a node is compromised, the secret material contained within is completely exposed and usable by the attacker. In order to cope with such threat, a few trust frameworks have been proposed in the literature to detect bogus sensor data. This implies a trust evaluation of sensor data at acquisition and aggregation time: trust refers to the reliability and accuracy of sensed information and it is related to the quality of the delivered sensor data.

6.1 Sensor Node Failure Detection

Within a WSN, sensor nodes are prone to different kind of failures, such as crash, omission, timing, value and arbitrary failures (61). Crash and omission imply no response from the sensor to the data query. Timing refers to timeout during the processing a request. Value failure deals with delivering incorrect value due to malfunctioning or compromised sensor nodes. Finally, arbitrary failures include all the types of failures that cannot be classified in previously described categories. For example, Byzantine failures (38, 35) describe a type of arbitrary failures that are in general caused by a malicious service that not only behaves erroneously, but also fails to behave consistently when interacting with other services and applications.

In sensor node failure detection, we identify self-diagnosis (28) and group detection (36, 17, 25, 42) approaches. With self-diagnosis, each nodes detect its own failure, e.g., based on battery exhaustion. In group detection, each node in the same area is supposed to deliver a similar information. A good example is temperature measurement in a room. Let us assume a WSN application to measure the temperature in a room: taking the average value provided by different thermometers in the same room makes it possible to resist attacks and to produce sensor data which is potentially more trustworthy as the number of contributors increases (37). Such a naive approach however raises the following issue: if the temperature values collected are $(10; 10; 11; 50)$, we get an average of over 20. It is obvious that the last value is a wrong one, and the reported temperature should be 10. The usage of an appropriate statistical method, e.g., median, allows us to detect that 50 is an outlier, and labels the sensor node delivering this value as unreliable. Nevertheless, this approach requires a large number of sensor nodes producing the same type of measurements.

6.2 Reputation System

Trustworthiness is often described as the expectation of cooperative behavior (22). Its evaluation is usually based on previous experiences with the same party. Thus an entity can establish trust in its communication partner based on the latter's reputation (58). The mathematical foundations for reputation management are rooted in statistics and probability (55). Reputation is defined as the perception that an entity has of another's intentions. Furthermore, reputation is based on a collection of evidence of good and bad behavior undertaken by other entities.

In (22), the authors integrate tools from different domains such as economics, statistics, data analysis and cryptography in order to establish trustworthiness of sensor nodes. This approach capitalizes on Bayesian formulation of reputation representation, updates, integration and trust evolution. The authors propose a Reputation based Framework for Sensor Network (RFSN), which can cope with bad mouthing and ballot stuffing attacks. For the former, the authors ignore all bad reputation information about others nodes, and keep only the good reputation information. For the latter, the authors propose to integrate the reputation on a node when updating its own reputation information about the other nodes. Thus in this approach, only good behaving nodes can get access to others nodes information.

In (9), the authors also propose a reputation system based on Bayesian approach. They clearly distinguish the reputation from trust in sensor nodes. The former represents the opinion formed by a node on another node in a sensor network. The latter represents the opinion formed by a node about how honest another node is in the reputation system. In this approach, each node is in charge of maintaining its reputation and trust rating on the node of its interest (e.g. the ones that it is interacting with). In addition, reputation systems (8, 43), originally designed for ad-hoc networks, are hardly applicable to WSN due to resource restriction on sensor nodes.

In all those approaches, authentication of sensor nodes is required. In order to bind a reputation to a sensor node, each node has to authenticate itself. Moreover, this type of approach does not propose any solution regarding the determination of reputation for the first interactions, when introducing a new node in the sensor network. Finally, approaches based on reputation system are time-expensive, since they require a lot of interaction between sensor nodes before establishing a stable trust relationship.

6.3 Trust Based Framework

We distinguish between reputation and trust. Reputation is based on past experiences with a given entity, whereas trust is not restricted to this. Trust enables to encompass objective and subjective characteristics on an entity. Reputation is part of the subjective characteristics which permits to determine trust, but not the only one. The goal of trust based framework for wireless sensor networks is to establish trust in all sensor nodes based on the expectation that they will deliver non-compromised data.

In (1), authors propose a trust framework for non-critical sensor network. Their approach consists of distributing, in clear text, cryptographic material for an node to another. Any nodes broadcasts his key material by increasing smoothly his communication signal strength. On contact, he nodes forwards its key material in clear text to the another node. The authors discuss the economic factors that would prevent any attacker from capturing key material. The attacker thus would have to deploy a lot of malicious nodes in order to increase his chance to recover cryptographic material.

In (46), the authors propose a trust- and clustering-based framework based on public key authentication for mobile ad hoc wireless networks. They define a trust model where each node monitors and rates each other with quantitative trust values. This trust model is totally decentralised and does not involve any trusted third party. In this approach, a chain of trust, similar to PGP, is established between nodes (49). Any node can sign another node's public key with its own private key. The authors developed a trust- and clustering-based public key authentication mechanism supported by new security operations on public key certification, update of a trust table. The goal is thus to discover and isolate dishonest nodes. Nevertheless those types of approach are still bounded by resource constraints on sensor nodes.

In (63), the authors propose a trust based framework for secure aggregation in wireless sensor network based on Bayesian model and beta distribution probability. They first evaluate trust in individual sensor nodes based on Kullback-Leibler (KL) distance or relative entropy. The idea is to calculate the distance between an ideal node behavior and the actual node behavior. In this case, the KL distance is the measure of the differences between two probability distributions: from a probability distribution P to an arbitrary probability distribution Q, with P is the real value and Q is the acquired sensor data. The authors assign a confidence value to aggregated sensor data. The opinion notion used in this approach finds its roots in subjective logic theory (32), in order to represent the uncertainty on the aggregation. Based on sensor data confidence, the framework computes an opinion which encompasses belief and uncertainty on the aggregation of sensor data by means of the consensus operator (32). Nevertheless, this approach is still time-consuming for establishing a stable reputation on sensor nodes. The reputation on a node, based the inverse square of its KL-distance, suffers from severe oscillation for the first reputation evaluations.

7 Conclusions

In-network data processing can potentially bring important benefits to wireless sensor networks. Computation is typically much less energy consuming than communication, so the additional computational overhead can be well justified by the reduced data transfer. The distributed processing, and possibility of aggregation or even partial reasoning about sensed data, could also enable WSNs to provide more complex services to application layer, and not only data gathering functionality. However before the concept of such intelligent sensor network becomes the reality, many technical challenges have to be addressed. In particular, we need to design and implement secure, yet very efficient and cost-effective, data aggregation mechanisms. Very promising results have been recently achieved in this area based on advanced cryptographic concepts, such as privacy homomorphisms, bilinear pairings, and elliptic curve cryptography. Another important issue is related to assessment of trustworthiness and reliability of the data provided by WSNs, especially when this data is pre-processed in the network and received by the application in an aggregated form. Several different approaches have been proposed to this problem, e.g. based on subjective logic.

Despite of potentially great importance and very interesting theoretical and practical challenges, the topic of secure in-network data processing in wireless sensor networks have received until recently much less attention than, e.g., secure routing or key management. Therefore, despite of many interesting initial results, the security questions related to distributed data processing in WSN remain largely open, and in our opinion constitute an interesting area for further research.

Acknowledgment

This work is partially financed by the European Commission under the Framework 6 IST Project *Wirelessly Accessible Sensor Populations (WASP)*.

References

[1] Ross Anderson, Haowen Chan, and Adrian Perrig. Key infection: Smart trust for smart dust. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, pages 206–215, Washington, DC, USA, 2004. IEEE Computer Society.

[2] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede. Low-cost elliptic curve cryptography for wireless sensor networks. In *Proceedings of the Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, 2006.

[3] Erik-Oliver Blaß and Martina Zitterbart. Towards acceptable public-key encryption in sensor networks. In *Proceedings of the Second International Workshop on Ubiquitous Computing (ACM SIGMIS)*, pages 88–93, 2005.

[4] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. A survey of two signature aggregation techniques. *CryptoBytes*, 6(2), 2003.

[5] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proceedings of the Second Theory of Cryptography Conference (TCC)*, pages 325–341, 2005.

[6] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of Theory of Cryptography Conference (TCC)*, pages 535–554, 2007.

[7] Ernest F. Brickell and Yacov Yacobi. On privacy homomorphisms (extended abstract). In *Proceedings of the EUROCRYPT*, pages 117–125, 1987.

[8] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks. In *Proceedings of the IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002.

[9] Sonja Buchegger and Jean-Yves Le Boudec. A robust reputation system for P2P and mobile ad-hoc networks. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[10] BusinessWeek. 21 ideas for the 21st century. Business Week, pp 78-167, 1999.

[11] D. W. Carman, P. S. Kruus, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical Report 00-010, Network Associates Inc., 2000.

[12] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. In *Proceedings of the Second Annual International Conference onMobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, pages 109–117, 2005.

[13] Certicom. Certicom announces elliptic curve cryptosystem challenge winner (press release), 1997. http://www.certicom.com.

[14] Haowen Chan, Adrian Perrig, and Dawn Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 278–287, 2006.

[15] Jung Hee Cheon, Woo-Hwan Kim, and Hyun Soo Nam. Known-plaintext cryptanalysis of the Domingo-Ferrer algebraic privacy homomorphism scheme. *Inf. Process. Lett.*, 97(3):118–123, 2006.

[16] Chee-Yee Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[17] M. Ding, D. Chen, K. Xing, and X. Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[18] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *Proceedings of the 5th International Conference on Information Security (ISC)*, pages 471–483, 2002.

[19] Elena Fasolo, Michele Rossi, Jörg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: A survey. *IEEE Communication Magazine*, April 2007.

[20] Gerhard Frey, Michael Müller, and Hans-Georg Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.

[21] Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the Tate pairing. In *ANTS-V: Proceedings of the 5th International Symposium on Algorithmic Number Theory*, pages 324–337, London, UK, 2002. Springer-Verlag.

[22] Saurabh Ganeriwal and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 66–77, New York, NY, USA, 2004. ACM.

[23] Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002.

[24] Johann Großschädl. TinySA: A security architecture for wireless sensor networks (extended abstract). In *Proceedings of the 2nd International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. ACM Press, 2006.

[25] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[26] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 119–132, 2004.

[27] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2003.

[28] S. Harte and A. Rahman. Fault tolerance in sensor networks using self-diagnosing sensor nodes. In *IEEE International Workshop on Intelligent Enviromnent*, 2005.

[29] Alireza Hodjat and Ingrid Verbauwhede. The energy cost of secrets in ad-hoc networks (short paper). In *Proceedings of the IEEE CAS Workshop on Wireless Communication and Networking*, 2002.

[30] R. Ismail and A. Josang. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[31] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In *CT-RSA '02: Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, pages 244–262, London, UK, 2002. Springer-Verlag.

[32] Audun Jøsang. A logic for uncertain probabilities. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 9(3):279–311, 2001.

[33] T. Kleinjung. Discrete logarithms in GF(p) — 160 digits. Nabble - Number Theory forum and mailing list archive. http://www.nabble.com/Discrete-logarithms-in-GF(p)—-160-digits-t3175622.html.

[34] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[35] Chiu-Yuen Koo. Broadcast in radio networks tolerating Byzantine adversarial behavior. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 275–282, New York, NY, USA, 2004. ACM Press.

[36] B. Krishnamachari and S. Iyengar. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Transactions on Computers*, 53(3):241–250, 2004.

[37] Sven Lachmund, Thomas Walter, Laurent Bussard, Laurent Gomez, and Eddy Olk. Context-aware access control. In *Proceedings of the 3rd Annual International Con-*

ference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS), 2006.

[38] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[39] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *J. Cryptology*, 14(4):255–293, 2001.

[40] Yonghe Liu and Sajal Das. Information-intensive wireless sensor networks: potential and challenges. *IEEE Communications Magazine*, 44(11):142–147, 2006.

[41] A. Mahimkar and T. Rappaport. Securedav: A secure data aggregation and verification protocol for sensor network. In *Proceedings of the IEEE Global Telecommunications Conference*, 2004.

[42] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000.

[43] Pietro Michiardi and Refik Molva. CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the Communication and Multimedia Security Conference (CMS)*, 2002.

[44] Victor S. Miller. Use of elliptic curves in cryptography. In *Proceedings of the Advances in Cryptology - CRYPTO*, pages 417–426, 1985.

[45] Refik Molva and Gene Tsudik. Secret sets and applications. *Inf. Process. Lett.*, 65(1):47–55, 1998.

[46] Edith C. H. Ngai and Michael R. Lyu. Trust- and clustering-based authentication services in mobile ad hoc networks. In *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, pages 582–587, Washington, DC, USA, 2004. IEEE Computer Society.

[47] Leonardo B. Oliveira, Diego Aranha, Eduardo Morais, Felipe Daguano, Julio Lo'pez, and Ricardo Dahab. TinyTate: Computing the Tate pairing in resource-constrained sensor nodes. In *Proceedings of the Sixth IEEE International Symposium on Network Computing and Applications (NCA)*, 2007.

[48] Suna Melek Önen and Refik Molva. Secure data aggregation with multiple encryption. In *Proceedings of the European Wireless Sensor Networks Conference (EWSN)*, 2007.

[49] OpenPGP Alliance. Open PGP. http://www.openpgp.org/.

[50] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the EUROCRYPT*, pages 223–238, 1999.

[51] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: Security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, 2002.

[52] S. Peter, K. Piotrowski, and P. Langendoerfer. On concealed data aggregation for wireless sensor networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference*, Jan. 2007.

[53] R. Rajagopalan and P.K. Varshney. Data-aggregation techniques in sensor networks: a survey. *Communications Surveys & Tutorials, IEEE*, 2006.

[54] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In R. DeMillo, D. Dobkin, A. Jones, and R. Lipton, editors, *Foundations of Secure Computation*, pages 169–180. Academic Press, 1978.

[55] P. Robinson and M. Beigl. Trust context spaces: An infrastructure for pervasive security. In *Proceedings of the First International Conference on Security in Pervasive Computing*, 2003.

[56] Wade Roush. 10 emerging technologies that will change the world. *Technology Review*, 106(2), 2003.

[57] Yingpeng Sang, Hong Shen, Yasushi Inoguchi, Yasuo Tan, and Naixue Xiong. Secure data aggregation in wireless sensor networks: A survey. In *Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 315–320, 2006.

[58] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, NJ, 1976.

[59] J. She and J. Yeow. Nanotechnology-enabled wireless sensor networks: From a device perspective. *IEEE Sensors Journal*, 6(5):1331–1339, 2006.

[60] Abdullatif Shikfa. Bilinear pairings over elliptic curves. Master's thesis, Ecole doctorale STIC de Nice Sophia-Antipolis, June 2005.

[61] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[62] H. Wang, B. Sheng, and Q. Li. Elliptic curve cryptography-based access control in sensor networks. *International Journal of Security and Networks*, 1(3-4):127–137, 2006.

[63] Wei Zhang, Sajal Das, and Yonghe Liu. A trust based framework for secure data aggregation on wireless sensor networks. In *Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 60–69, 2006.

Author Biographies

Alessandro Sorniotti (alessandro.sorniotti@sap.com) is a Research Associate at SAP Labs France in the Department of Security and Trust since 2007, and also a PhD candidate at Ecole Nationale Supérieure des Télécommunications de Paris, France since 2007. He received his MSc in Computer Science from Politecnico di Torino, Italy in 2007. He also holds a Master in Networking and Distributed System from Ecole Polytechnique de l'Université de Nice-Sophia Antipolis, France in 2006. His current research interests lie in the security of Wireless Sensor Networks.

Laurent Gomez (laurent.gomez@sap.com) is a senior researcher for SAP Research in the Department of Security

and Trust since 2001. He received his engineer degree in computer science from Ecole Superieure en Sciences Informatiques, Sophia Antipolis, France in 1999. His current research interests lie in the area of secure integration of Wireless Sensor Networks into business application and trusted in-network data aggregation and reasoning.

Konrad Wrona (kwrona@ieee.org) Dr.-Ing. Konrad Wrona is currently a Principal Investigator at SAP Research Lab in Sophia Antipolis, France. He has over ten years of work experience in an industrial (SAP Research and Ericsson Research) and in an academic (RWTH Aachen University, Media Lab Europe, and Rutgers University) research and development environment. He has earned his M.Eng. in Telecommunications from Warsaw University of Technology, Poland in 1998, and his Ph.D. in Electrical Engineering from RWTH Aachen University, Germany in 2005. He is an author and a co-author of over twenty publications, as well as a co-inventor of several patents. The areas of his professional interests include security in communication networks, wireless and mobile applications, distributed systems, applications of sensor networks, and electronic commerce.

Lorenzo Odorico is an Intern at SAP Labs France in the Department of Security and Trust, where he is currently writing his Master Thesis. He received his BSc in Computer Science from Politecnico di Torino, Italy in 2005, and he expects his Master in Networking and Distributed System from Ecole Polytechnique de l'Université de Nice-Sophia Antipolis, France in 2007, and his MSc in Computer Science from Politecnico di Torino, Italy in 2008. His current research interests lie in the security of Wireless Sensor Networks.