CDA: Concealed Data Aggregation in Wireless Sensor Networks

Joao Girao
NEC Europe Ltd.
69115 Heidelberg, Germany
joao.girao@netlab.nec.de

Markus Schneider
Fraunhofer-SIT
64293 Darmstadt, Germany
schneider@sit.fhg.de

Dirk Westhoff
NEC Europe Ltd.
69115 Heidelberg, Germany
dirk.westhoff@netlab.nec.de

ABSTRACT

End-to-end encryption for wireless sensor networks is a challenging problem. To save the overall energy resources of the network it is agreed that sensed data need to be consolidated and aggregated on their way to the final destination. For such circumstances we present an approach that conceals sensed and aggregated data end-to-end. Even the aggregating intermediate nodes are not enabled to read the sensed plaintext data. We apply a particular class of encryption transformation and exemplary discuss the approach on the basis of two aggregation functions. We show their appliance in hierarchical aggregator topologies and use actual implementation to show that the approach is feasible and frequently even more energy efficient than hop-by-hop encryption addressing a much weaker attacker model.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: General—*Network protocols, Security and protection*; E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security, Algorithms

Keywords

Wireless sensor networks, Data encryption, Data aggregation, Energy consumption, Privacy Homomorphism

1. PROBLEM STATEMENT

Wireless sensor networks (WSN) are a particular class of ad hoc networks that attract more and more attention both in academia and industry. The sensor nodes themselves are preferably cost-cheap and tiny consisting of a) application specific sensors, b) a wireless transceiver, c) a simple processor, and d) an energy unit which may be battery or solar driven. Such sensor nodes are envisioned to be spread out over a geographical area to form in an indeed self-organizing manner a multihop network. Most frequently such WSNs are stationary, although mobile WSNs are also conceivable. Potential applications for WSNs - beside military ones - can be found in monitoring environmental data with the objective to understand complex and geographical wide spread interdependencies of nature. Examples are the detection of fire in huge forest areas, the monitoring of wildlife animals' movement patterns, or the incremental shift of snow and rocks in the alpine mountains. Further applications for wireless sensor networks are envisioned to be on the biomedical sector. Even monitoring the health status of cattle stocks on farms may be supported by a WSN.

As aforementioned, one major application scenario for a WSN is to monitor environmental data and to transmit them to a central point. Here the data are analyzed and eventually serve to initiate some action. Analysis in most scenarios presumes computation of an optimum e.g. the minimum or maximum, the computation of the average, or the detection of some movement pattern. Precomputation of these operations may be either fulfilled at a central point or by the network itself. The latter is beneficial in order to reduce the amount of data to be transmitted over wireless connections. Since the energy consumption increases linearly with the amount of transmitted data, an aggregation approach helps increasing the WSN's overall lifetime. Another way that helps saving energy is to maintain only a connected backbone whereas nodes that perform no forwarding task in this backbone persist in sleep mode until they are activated by nodes from the backbone.

Within the considered aggregation scenario for stationary WSNs one needs to logically separate between sensor nodes $S_1, \ldots, S_n$, aggregator nodes $A_1, \ldots, A_l$ and the sink node R, which we assume to initiate the monitoring and data collecting process. Aggregator nodes belong to the backbone whereas sensor nodes persist in sleep mode until the sink node initiates a process which requires a subset of them to contribute. For simplicity in this work we assume the nodes belonging to the backbone to be fix. In principle they should be periodically changed. The sink node may either be the connection to the fixed network or the end point for the data collection process. It is assumed to be much more powerful than those nodes which perform the sensing and the data aggregation. Note that in general the distance of a sensor node to the nearest aggregation node is not necessarily one hop. On the contrary, even sensor nodes may contribute to the aggregation task.

In the remainder of this work we denote s_i as the spatio-temporal sensed datum of S_i during a time interval $[t, t+\Delta t]$ in the environmental region (x_i, y_i, r). Here, the coordinates x_i, y_i denote the position of S_i. The radius r denotes the maximum distance relative to x_i, y_i from where the sensed environmental data stems from. More precisely, s_i is the local optimum of S_i over a well defined time interval and monitoring cube with respect to the requested aggregation function. Typically r is much smaller than the transmission range r' $(r \ll r')$ of a node which we expect to be homogeneous for all nodes and in the range of 5m - 30m.

Ideally the highest entropy on the monitored system status is achieved by sending the s_i of each S_i to the sink node via As with a degenerated aggregation function that results in a simple forwarding process. This may be the transmission of a single message $s_1||\dots||s_n$ or of n sequential messages $s_1, \dots, s_n$. However, in order to maximize the lifetime of the WSN, it is wise to compress these messages by not falling below the entropy necessary for the requested information. It is expected that at time $t + \Delta t$ an aggregator node A computes on base of the received sensed values the aggregator function $f : \{0,1\}^k \times \dots \times \{0,1\}^k \to \{0,1\}^{k+l}$ with $y = f(s_1, \dots, s_n)$. To reduce traffic overhead, and with respect to an energy efficient approach, only the aggregation value y is subsequently transmitted to the sink node R.

1.1 Background

Contrary to the sink node, the sensor nodes and the aggregation nodes have a very restricted capacity and computation power. We even cannot assume that devices contain a tamper-resistant hardware unit. Thus, when applying WSNs to do data aggregation in a probably hostile or faulty environment, one has to take into account that a subset of nodes has been corrupted and misbehaves. Aggregator nodes, which are envisioned to collect data from sensoring nodes in their neighborhod, are a particular reasonable aim for an attacker since here sensed information from a whole region is available and will be consolidated. In this work we restrict ourselves assuming an attacker who tries to spy out consolidated sensor data of a whole region. We assume that the attacker is not in the position to simply measure these data or to corrupt each of the involved sensor nodes since he is physically limited to one place. Thus the attacker will corrupt the aggregator node with the aim to read all ingoing sensed data. The contribution of this work is the provision of a data aggregation approach for WSNs providing end-to-end concealment on the sensed data as well as on the aggregated data between the sensors and the sink. In particular this means that even the aggregator node is not able to read the sensed data on which it performs the aggregation function. Note that, although these are mandatory issues, we will not provide integrity and plausibility of the sensed data. For the first, Bohge and Trappe in [2] provided on base of Perrig's TESLA [6] an authentication framework for hierarchical ad hoc sensor networks which to some extend may support our contribution. Another security architecture for in-network processing in WSNs is described in [4]. Further candidates for the authentication of sensor nodes and sensed data may be μTESLA [7], ZCK [13] or even Guy Fawkes [1]. Approaches that deal with plausible data aggregation are from Boulis et al. [3] and SIA [10] from Przydatek et al. Both approaches focus on the efficiency-accuracy trade-off for computing plausible aggregation data. The latter also takes a subset of corrupted nodes into account. However, the purpose of this at hand work is the provision of an approach against passive attacks, e.g. to conceal sensed data on their way from the monitoring source to the sink node.

The rest of the paper is organized as follows: Section 2 introduces a particular class of encryption transformations, namely privacy homomorphisms. In Section 3 we describe a privacy homomorphism introduced by Domingo-Ferrer. In Section 4 this approach is applied to the problem of concealed data aggregation in WSNs. Sections 5 and 6 give a proof of concept according to the specific aggregation functions *average* and *detect moving entity.* Parameter settings are discussed in Section 7. In Section 8 we discuss the concealed data aggregation approach for appliance in hierarchical WSN topologies before in Section 9 we exemplary show how it fits to the requirements of a particular destination platform. Finally, Section 10 contains our initial conclusions from this research.

2. PRIVACY HOMOMORPHISMS

We introduce a particular class of encryption transformations which we show are valuable to overcome the problem of concealed data aggregation in WSNs.

A privacy homomorphism (PH) is an encryption transformation that allows direct computation on encrypted data. Let Q and R denote two rings, $+$ denote addition and $\times$ denote multiplication on both. Let K be the keyspace. We denote an encryption transformation $E : K \times Q \to R$ and the corresponding decryption transformation $D : K \times R \to Q$. Given $a, b \in Q$ and $k \in K$ we term

$$a + b = D_k\big(E_k(a) + E_k(b)\big) \tag{1}$$

additively homomorphic and

$$a \times b = D_k\big(E_k(a) \times E_k(b)\big) \tag{2}$$

multiplicatively homomorphic.

First work on PHs was done in a seminal paper by Rivest et al. [8]. Generally, the more operands a PH supports the more computation intensive the transformations E and D are. E.g. RSA is a multiplicative PH. In [9] Domingo-Ferrer presented an additive and multiplicative PH. Although in [12] Wagner showed that the proposed PH is insecure for some major parameter settings we believe that for the WSN data aggregation scenario this scheme is still reasonable secure. We argue that sensed and aggregated data are highly transient and thus have value for an attacker only for a very limited period of time, say one day or even less. For some specific parameter settings the PH proposed by Domingo-Ferrer will still conceal information for such a period. Nevertheless, generally it is provable that an additively PH is insecure against chosen plaintext attacks.

3. AN ADDITIVE AND MULTIPLICATIVE PH

We describe the parameter settings, encryption transformation and decryption transformation of the PH proposed by Domingo-Ferrer.

Settings: The public parameters are

- a positive integer $d \geq 2$ and a large integer g. It is pointed out that g should have many small divisors and at the same time there should be many integers less than g that can be inverted modulo g.

The secret key is

- $k = (r, g')$. The value $r \in \mathbb{Z}_g$ is chosen such that, first, $r^{-1} \bmod g$ exists, and second, $log_{g'} g$ is an integer with small g'.

The set of cleartext is $\mathbb{Z}_{g'}$ and the set of ciphertext is $(\mathbb{Z}_g)^d$. Encryption and decryption transformation work as follows:

Encryption: Randomly split $a \in \mathbb{Z}_{g'}$ into a secret $a_{.1}, \dots, a_{.d}$ such that $a = \sum_{j=1}^{d} a_{.j} \bmod g'$ and $a_{.j} \in \mathbb{Z}_{g'}$. Compute

$$E_k(a) = (a_{.1} r \bmod g, a_{.2} r^2 \bmod g, \dots, a_{.d} r^d \bmod g) \quad (3)$$

Decryption: Compute the j-th coordinate by $r^{-j} \bmod g$ to retrieve $a_{.j} \bmod g$. To get a compute

$$D_k\big(E_k(a)\big) = \sum_{j=1}^{d} a_{.j} \bmod g'. \quad (4)$$

The ciphertext operation $+$ is done componentwise. For the ciphertext operation $\times$ all terms are cross-multiplied in $\mathbb{Z}_g$, with the d_1-degree term by a d_2-degree term yielding a $d_1 + d_2$-degree term. Terms having the same degree are added up.

4. DATA AGGREGATION WITH PH

In presence of the previously motivated and introduced passive attacker model we propose applying Domingo-Ferrer's approach to conceal the process of data aggregation in a WSN: Sensors S_1 to S_n encrypt their data s_1 to s_n resulting in $s'_1 = E_{(r,g')}(s_1)$ to $s'_n = E_{(r,g')}(s_n)$ before transmitting data to the A. Then, A operates on the encrypted data and computes $y' = f(s'_1, \dots, s'_n)$. Subsequently, the aggregator A transmits y' to the R which decrypts the y' and derives the accumulated data $y = D_{(r,g')}(y')$. Figure 1 illustrates the approach.

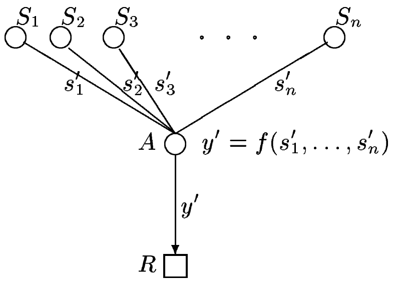

Figure 1: Concealed Data Aggregation for WSNs with PH.

More precisely, the concealed data aggregation for a WSN with Domingo-Ferrer's PH works as follows:

- We consider (r, g') to be known to $S_1, \dots, S_n$ and at the R. The values d and g are public and thus can also be known to A. The aggregation function with its additive and/or multiplicative operations is also public and known to A and to $S_1, \dots, S_n$.
- At S_i with $1 \leq i \leq n$: Split $s_i \in \mathbb{Z}_{g'}$ into a secret $s_{i.1}, \dots, s_{i.d}$ such that $s_i = \sum_{j=1}^{d} s_{i.j} \bmod g'$ and $s_{i.j} \in \mathbb{Z}_g$. Compute $s'_i = E_{(r,g')}(s_i) = (s_{i.1} r \bmod g$, $\mathrm{s}_{i.2} r^2 \bmod g, \cdots, a_{i.d} r^d \bmod g)$ and transmit s'_i to the A.
- At A: Compute on base of the additive and multiplicative homomorphic operations $+$ and $\times$ the aggregation function $y' = f(s'_1, \dots, s'_n)$ and transmit y' to the R.
- At R: Compute the scalar product of the j-th coordinate by $r^{-j} \bmod g$ to retrieve $s_{i.j} \bmod g$. Subsequently compute $y = D_{(r,g')}(y') = \sum_{j=1}^{d} s_{i.j} \bmod g'$.

Next, we exemplarily describe the approach for the aggregation functions *average* and *detect movement pattern*. One reason for choosing these aggregation functions is their difference with respect to a nested arrangement. In Section 8 we will discuss the consequence of this feature in more detail. Note that for both chosen aggregation functions the approach only needs to apply the additive operation, thus causing only relatively moderate data overhead.

5. AVERAGE COMPUTATION

Assume $n = 5$ sensors which monitor environmental data, say they are monitoring data $(s_1, s_2, s_3, s_4, s_5) = (1, 2, 1, 0, 1)$. For illustration we choose unrealistic small values $d = 2$ and a public modulus $g = 28$. The public aggregation function *average* is $f(s_1, \ldots, s_n) = \frac{\sum_{i=1}^n s_i}{n}$. Let $r = 3$ and $g' = 7$ be the secret key and $n = 5$ known to R.

S_i with $1 \leq i \leq 5$ e.g. compute

$$\begin{array}{rcl} s_1' & = & E_{(3,7)}(1) = E_{(3,7)}(4,4) = (12,8) \\ s_2' & = & E_{(3,7)}(2) = E_{(3,7)}(7,2) = (21,18) \\ s_3' & = & E_{(3,7)}(1) = E_{(3,7)}(6,2) = (18,18) \\ s_4' & = & E_{(3,7)}(0) = E_{(3,7)}(3,4) = (9,8) \\ s_5' & = & E_{(3,7)}(1) = E_{(3,7)}(3,12) = (9,24) \end{array} \quad (5)$$

and transmit

$$\begin{array}{rcl} S_1 \rightarrow A & : & (12,8) \\ S_2 \rightarrow A & : & (21,18) \\ S_3 \rightarrow A & : & (18,18) \\ S_4 \rightarrow A & : & (9,8) \\ S_5 \rightarrow A & : & (9,24). \end{array} \quad (6)$$

A computes

$$\begin{array}{rcl} y' & = & \sum_{i=1}^{n} E_{(3,7)}(s_i') \\ & = & (12 + 21 + 18 + 9 + 9 \bmod 28, \\ & & 8 + 18 + 18 + 8 + 24 \bmod 28) \\ & = & (13, 20) \end{array} \quad (7)$$

and transmits

$$A \rightarrow R : (13, 20). \quad (8)$$

R computes

$$\begin{array}{rcl} y & = & \frac{D_{(3,7)}(y')}{n} \\ & = & \frac{(13 \times r^{-1} \bmod 28, 20 \times r^{-2} \bmod 28) \bmod 7}{5} \\ & = & \frac{(13 \times 19 \bmod 28, 20 \times 19^2 \bmod 28) \bmod 7}{5} \\ & = & \frac{(23,24) \bmod 7}{5} \\ & = & \frac{5}{5} \\ & = & 1. \end{array} \quad (9)$$

Verification: $\frac{\sum_{i=1}^n s_i}{n} = \frac{1+2+1+0+1}{5} = 1$ □

6. MOVEMENT DETECTION

Next, we describe how Domingo-Ferrer's PH can be applied to the problem of a concealed movement detection function. The movement pattern of an entity that crosses the region covered by the WSN shall be communicated in a concealed manner. Before describing the approach for the more general sensor topology *field* we present the approach for the sensor topologies *chain* and *circle*.

6.1 Movement Detection for Chains and Circles

Again assume $n = 5$ sensors now monitoring movement patterns, say $(s_5, s_4, s_3, s_2, s_1) = (0, 0, 0, 1, 1)$.

A 0-bit transmitted by S_i and finally understood at R as the $(n + 1 - i)$-th position in the aforementioned binary tuple means *monitoring no moving entity* within region (x_i, y_i, r), whereas transmitting a 1-bit means *monitoring moving entity* within region (x_i, y_i, r). Assume that sensors S_1 to S_n are aware of their relative positions to each other and in addition the sensors know n. Also assume that R is aware of the S_is positions (x_i, y_i, r). W.l.o.g. S_i is a direct neighbor to S_{i-1} and S_{i+1}. More precisely, the sensors topology is a chain, or, if S_1 and S_n are also direct neighbors, the sensors establish a circle. From the 5-tuple above noted one can infer that entities have moved from (x_2, y_2, r) to (x_1, y_1, r) (or vice versa).

In principle a sensor which has monitored no movement needs to send the value $s_i = 0 \in \mathbb{Z}_{g'}$ to the A whereas in case of a movement detection the S_i sends $s_i = 2^{i-1} \in \mathbb{Z}_{g'}$ to the A. Although the scheme of Domingo Ferrer ensures varying ciphers if the same plaintext is encrypted several times, we introduce a nonce which also gives freshness and in addition virtually increases the cleartext space. Since with $|Q| = 2$ the set of cleartext is very limited we propose that R reveals with each aggregation request some additional value $l \in \mathbb{Z}_{g'}$ to the S_is. The S_is add l to the s_is and transmit the result to the A. Thus we extended the set of cleartext to $|Q| = g'$ at the cost of some pre-established additional group key between the S_is and the R. This we do although the PH from Domingo-Ferrer itself guarantees a changing ciphertext if the same plaintext needs to be encrypted several times. Nevertheless our extension helps reducing the probability for a chosen plaintext attack since in addtion it increases the plaintext space.

For a more detailed description let $g = 56$ and $(r, g') = (3, 14)$. Again we choose $d = 2$. The R chooses $l = 2$ and broadcasts its cipher concatenated with the aggregation function to the WSN. We exemplarily describe the computation at sensor nodes S_2 and S_3 for the above introduced sensoring tuple: Since the S_2 monitored some movement and with the knowledge of n and its own position in the chain it has to translate $2^{i-1} = 2^1$ to the binary n-tuple $(0, 0, 0, 1, 0)$. Thus, the cleartext representation of $(0, 0, 0, 1, 0)$ is $2 \in \mathbb{Z}_{g'}$ which needs to be added with $l = 2 \in \mathbb{Z}_{g'}$ resulting in $s_2 = 4$. On the other hand, since the S_3 has not observed any movement it computes $s_3 = 0 + l = 2$. Subsequently, the S_is apply Domingo-Ferrer's encryption transformation, e.g. the

S_1 to S_5 compute:

$$\begin{aligned} s'_1 &= E_{(3,14)}(3) = E_{(3,14)}(2,1) = (6,9) \\ s'_2 &= E_{(3,14)}(4) = E_{(3,14)}(11,7) = (33,7) \\ s'_3 &= E_{(3,14)}(2) = E_{(3,14)}(2,18) = (6,28) \\ s'_4 &= E_{(3,14)}(2) = E_{(3,14)}(1,1) = (3,9) \\ s'_5 &= E_{(3,14)}(2) = E_{(3,14)}(13,17) = (39,41) \qquad (10) \end{aligned}$$

and transmit the results to the A. The A computes

$$\begin{aligned} y' &= \sum_{i=1}^{n} E_{(3,14)}(s'_i) \\ &= (6+33+6+3+39 \bmod 56, \\ &\quad 9+7+28+9+41 \bmod 56) \\ &= (31,38) \qquad (11) \end{aligned}$$

before transmitting it to the R. Subsequently, the R computes

$$\begin{aligned} y &= D_{(3,14)}(y') \\ &= (31 \times 19 \bmod 56, 38 \times 19^2 \bmod 56) \bmod 14 \\ &= (29,54) \bmod 14 \\ &= 13. \qquad (12) \end{aligned}$$

Finally, since $n = 5$, the R decreases five times the value $l = 2$ resulting in $13 - 10 = 3$.

Verification: $(3+4+2+2+2) - 5 \cdot 2 = 3$ □

The binary representation of the decrypted value is 3 and in the 5-tuple it is $(s_5, s_4, s_3, s_2, s_1) = (0,0,0,1,1)$. From this information the R can infer that an entity has moved from (x_1, y_1, r) to (x_2, y_2, r) or vice versa. Finally, note that applying Domingo-Ferrer's PH to the concealed movement detection mandatory needs to hold $g' \geq 2^{n-1} + n \cdot l$.

6.2 Movement Detection for Fields

The basic scheme for a concealed movement detection for a sensor chain or a circle is also extendable for a sensor field. Let $n \times m$ sensors be dimensioned on a rectangular field with $S_{i,j}$ and $1 \leq i \leq n$, $1 \leq j \leq m$ be direct neighbor of $S_{i,j+1}$, $S_{i,j-1}$, $S_{i+1,j}$ and $S_{i+1,j+1}$ and the R be aware of the n and the m. Each $S_{i,j}$ now transmits $s'_{i,j}$ meaning that the A receives $n \cdot m$ sensed and encrypted values. Figure 2 illustrates this. The A now behaves as if the topology would be a sensor chain of $n \cdot m$ sensors.

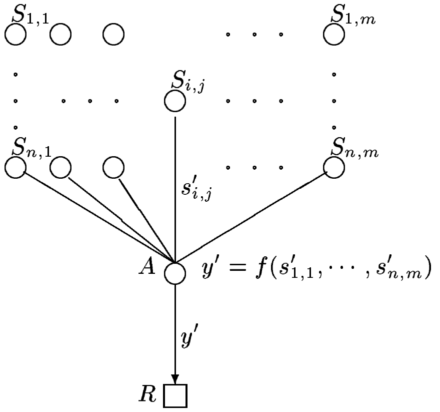

Figure 2: Concealed Movement Detection for WSN Fields with PH.

Under the assumption that $g' \geq 2^{n+m-1} + n \cdot m \cdot l$ it computes

$$y' = \sum_{i=1}^{n} \sum_{j=1}^{m} E_{(r,g')}(s'_{i,j}) \qquad (13)$$

and transmits the y' to the R. The R computes $y = D_{(r,g')}(y')$ and translates the $y \in \mathbb{Z}_{g'}$ in a binary $(m \cdot n)$-tuple. Subsequently it subdivides this tuple into m separate n-tuples. E.g. assume the R separated 3 tuples each of 5 elements

$$\begin{aligned} (s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}, s_{1,5}) &= (0,0,0,1,0) \\ (s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}, s_{2,5}) &= (0,0,1,0,0) \\ (s_{3,1}, s_{3,2}, s_{3,3}, s_{3,4}, s_{3,5}) &= (1,1,0,0,0) \qquad (14) \end{aligned}$$

it can infer that an entity has moved from $(x_{3,1}, y_{3,1}, r)$ via $(x_{3,2}, y_{3,2}, r)$ and $(x_{2,3}, y_{2,3}, r)$, to $(x_{1,4}, y_{1,4}, r)$ or vice versa. Note that although it needs to hold $g' \geq 2^{n+m-1} + n \cdot m \cdot l$ the size of y' is still only $|y'| = d \cdot |y|$ like for the chain or circle topology. This statement holds for aggregation functions based on additive operations.

7. PARAMETER DISCUSSION

Applying Domingo-Ferrer's PH to the WSN scenario we face three limiting factors. First, for purely additive aggregation functions, the size of the encrypted message increases factor d to the plaintext, e.g. $|y'| = d \cdot |y|$ and $|s'| = d \cdot |s|$ with $y' = f(s'_1, \ldots, s'_n)$ and $f : \{0,1\}^{d \cdot k} \times \ldots \times \{0,1\}^{d \cdot k} \rightarrow \{0,1\}^{d \cdot (k+l)}$. Thus, although when solely arguing from the security level one should choose a $d \gg 2$, by also considering the data overhead and the fluctual character of the sensed data we propose to limit d to a value in the range of $2-4$. The concrete value for d may vary with respect to the destination platform. In Section 9 we show that a d-times larger cipher will not result in the same ratio of transmission overhead. We do propose a relatively small d although in [12] Wagner points out that there is only one case where it is not known how to attack Domingo-Ferrer's PH, namely when g' is chosen to be hard to factor and the set of ciphertexts is restricted to $|R| < d$. We argue that, even under the obvious weakness of Domingo-Ferrer's PH of equivalent key pairs, the proposed ways to recover secrets g' and r are still much too time consuming for highly transient sensed values. Wagner proposes to collect a pool of known plaintexts and their encryptions to recover g'. Once g' is recovered he proposes to infer r either by exhaustive search, an attack based on linear algebra or an attack based on polynomial root finding.

Second, it needs to hold $g' > y$. This limitation is considerably influenced by i) the number of operands, namely the number of sensors n per aggregator node, and ii) $|Q|$ the number of elements in the set of cleartext. E.g., for additive homomorphic operations, if the cleartext set counts $|Q| = 256$ elements and the information of $n = 10$ sensors are bundled by the A then on average it needs to be $g' > 1280$. Here we assume the probability of occurrence of a sensed value to be equally distributed over Q. In this example and with proposed $d = 2$ the size of an encrypted message increases from $|s| = 1$ byte plaintext to $|s'| = 2$ bytes ciphertext. The size of y' also doubles.

The third limiting factor for applying Domingo-Ferrer's PH to WSNs is the execution time at the nodes. In this Section we argue independent of a concrete destination platform but with respect to the used security parameter. We argue that the key generation phase and the configuration of d, g, and (r, g') is a setting which is performed by the manufacturer before the WSN is layed out. Execution times for this pre-configuration of the WSN are uncritical due to energy consumption and are not considered here. Execution times for an encryption transformation at the sensor nodes depend on the choice of d. We illustrate the influence of d on the number of costly operations in Table 1. For a varying d the necessary number of addition operations, multiplication operations and division operations are listed. Note, that they are independend of g, s' and r. In Table 1 we separate between operations at the S, the A and the R. Recall that operations at the R are uncritical since we envision the sink to be much more powerful than the other nodes.

Table 1: Computation effort for concealed data aggregation at sensor node, aggregator node, and sink node in terms of operations for varying d.

	encrypt (at S)			add (at A)			decrypt (at R)		
d	+	×	%	+	×	%	+	×	%
2	8	3	2	4	0	2	4	4	1
3	13	5	5	6	0	3	6	6	1
4	16	7	7	8	0	4	8	8	1
5	20	9	9	10	0	5	10	10	1
8	28	15	13	16	0	8	16	16	1
10	38	19	18	20	0	10	20	20	1

The numeric value of g defines the value space on which the above operations occur. If this value is too large, it may happen that they cannot be handled strictly by the processor. In the case of the Mica Motes, operands larger than 8 bits have to be handled through the use of special sofware routines. We therefore conclude that, should g be larger than 256, software operations have to assist the hardware instruction set - which consumes more clock cycles and power. Each time g's value requires another byte for its representation, the more complex the operation will be at software level. For the measurements depicted in Table 1, we set the value of g not larger than 4 bytes.

Summing it up: Since the d has influence on both, the data overhead and the execution times, we propose to use a moderate d, e.g. $d \leq 4$. Also the g should be used in a ballanced way to ensure on the on hand an approppriate level of security and on the other hand only moderate computation. We feel that $g \leq 2^{32}$ is an appropriate choice for the envisioned scenario.

8. APPLIANCE TO HIERACHICAL WSN TOPOLOGIES

Next, we discuss if the concealed data aggregation based on Domingo-Ferrer's PH is usable in a hierarchical manner. More precisely, we validate if an aggregator node can also process encrypted data from another aggregator node, which already did some data aggregation on a subset of sensed data by still processing an overall correct y. Independently of the PH's algebraic properties a hierarchical concealed data aggregation only holds if the aggregation function itself has the following characteristic:

$$f(s_1, \dots, s_n) = f\big(f(s_1, \dots, s_{h1}), \dots, f(s_{n-hl}, \dots, s_n)\big) \quad (15)$$

Clearly, the aggregation function sum fulfills this characteristic. On the other hand the aggregation function movement detection does not support this characteristic. However, the aggregation function average supports this property. This is true under the assumption that each partial aggregator A_j with $j = 1, \dots, l$ of the same hierachy level also knows h_j and the h_j for all l aggregator nodes from the same hierarchy have the same size h. Referring to Eq. (15) the aggregation function average then holds the above characteristic:

$$\frac{\sum_{i=1}^{n} s_i}{n} = \sum_{j=1}^{l} \Big(\frac{\sum_{i=1}^{h} s_i}{h}\Big) \quad (16)$$

Unfortunately, Domingo-Ferrer's PH does not provide division since the polynomials are a ring and not a field. We thus propose to modify Eq. (16) to

$$\frac{\sum_{i=1}^{n} s_i}{n} = \frac{\sum_{j=1}^{l} \sum_{i=1}^{h} s_i}{n} \quad (17)$$

to hold

$$\frac{\sum_{i=1}^{n} E_{(r,m')}(s_i)}{n} = \frac{\sum_{j=1}^{l} \sum_{i=1}^{h_j} E_{(r,m')}(s_i)}{n}. \qquad (18)$$

In this manner the aggregator nodes perform the aggregation function sum which fulfills Eq. (15) and subsequently at the sink node the received aggregation value is divided by n. As a side effect with this approach only the sink needs to know n. Note that also in a hierarchical aggregator scenario the encryption is only done at the leaves (sensoring nodes). Decryption is exclusevly done at the powerful sink node.

9. REAL WORLD CONSIDERATION

In this Section we present concrete measurements from our implementation and show how applying the PH for a concealed end-to-end encryption helps

- distribute the energy consumption over all nodes in a more ballanced way, and
- reduce the energy load in the backbone for a major class of WSN topologies.

Carefully distributing the energy consumption over the WSN is favorably since this reduces the risc of a disconnected WSN due to nodes with empty batteries. In fact, for maintaining a connected backbone of the WSN it is even preferable to perform energy consuming actions at the leaves while at the same time saving as much energy as possible in the backbone. Unfortunately, in presence of encryption protocols that work on a hop-by-hop basis, aggregator nodes are endangered to loose their energy much earlier than other nodes since sensed data need to be computed in plaintext at the aggregator node. Consequently, the aggregator node has to accomplish additional decryption and encryption operations before and/or after performing the aggregation function which unnecessarily reduces the battery of a node from the backbone. We will substantiate this statements by considering a homogeneous WSN, meaning that nodes have the same destination platform, they are equipped with the same battery, and they transmit data over the same range.

Crossbow's Mica Motes are one candidate for a destination platform. We evaluate the performance of our approach for the Mica2 Motes [5] and compare it to a simple hop-by-hop encryption with RC5 that is provided when using TinySec [11]. We consider the main operations in each of the approaches, namely addition, subtraction, multiplication and division (modular operation). Although they do not contemplate all processor instructions used in the algorithm's implementations, we believe these to be a significative sample for a comparison. We collected these values in a statistical form with a uniform variation on the data to be encrypted as well as on the keys generated for the operations. Note, that we do not aim at an absolute value study for our implementation but rather a comparative study that will allow us to make certain considerations on when Domingo Ferrer's PH (DF) is applicable.

For the encryption transformations of $RC5$ versus $DF_{d=2}$, $DF_{d=3}$, and $DF_{d=4}$ we measured execution times in terms of clock cycles for encryption and decryption of one byte plaintext data. Furthermore we measured the clock cycles for an addition of 10 plaintext operands each of one byte as well as clock cycles for an encrypted addition with Domingo Ferrer's PH. Due to the necessity of a random choice of r, the clock cycles for encryption with Domingo Ferrer's PH can only be given approximately. In Table 2 we thus list an average value from our measurements.

Table 2: Execution times for a Mica2 Mote in clock cycles [cc] for hop-by-hop based encryption based on TinSec's RC5 and end-to-end encryption based on Domingo-Ferrer's PH.

	encrypt [cc]	decrypt [cc]	add [cc]
	at $S_i, i = 1, \dots n$	at R	at A
RC5	236	236	4
DF$_{d=2}$	1951	2330	1452
DF$_{d=3}$	3481	3136	2178
DF$_{d=4}$	4277	3942	2904

At a first glance the above measurements clearly indicate our concerns: Encryption, decryption and also addition are by far more expensive comparing the clock cycles that are necessary to perform Domingo-Ferrer's PH with those necessary to perform RC5. Nevertheless the approach may still be reasonable with respect to the distribution of the overall energy consumption in the WSN. From the above values one can approximate that for $d = 2$ a WSN topology with more than six sensor nodes per aggregator node results in less computation overhead at the aggregator node than using hop-by-hop encryption based on RC5 ($1452 \leq (n+1) \times 236$). For example, assuming each aggregator node to be responsible for ten sensors the PH of Domingo Ferrer still takes 1452 clock cycles whereas clock cycles for applying hop-by-hop encryption are nearly twice as much. Although for $d = 3$ and $d = 4$ the break even shifts to nine respectively twelve nodes we believe that this is still a realistic bundle of sensor nodes per aggregator node. Unfortunately, this performance gain at the aggregator node comes at a performance loss at the sensor nodes due to costly encryption. We argue that for a homogeneous WSN with respect to the major objective to advantageously ballance the energy consumption this disadvantage is acceptable since the aggregator node clearly is the performance bottleneck when maintaining a connected WSN backbone. In fact the only restriction at the sensing nodes due to computation and energy consumption is that the consumed energy should not be dramatically higher than the one consumed at the aggregator nodes. To recall, sensor nodes persist a considerable period of their lifetime in sleep mode.

Furthermore, for a fair evaluation of a hop-by-hop encryption scheme and our end-to-end concealed data aggregation for WSNs we also need to consider the platform's radio stack. For the Mica Motes, a TinyOS (TOS) packet is preconfigured with a maximal size of 36 bytes, 29 bytes payload, 2 bytes CRC and some other information on address, type, group and length. Taking the TOS packet format into consideration a TinySec-AE encrypted TOS packet with sensed data of 1 byte and $|Q| = 256$ is of size 9 bytes [1] whereas the corresponding Domingo Ferrer PH encrypted TOS packet is of sizes 9 bytes up to 11 bytes (assuming either $d = 2 - 4$). Thus, the additional data overhead of the concealed data aggregation compared to an RC5 protected data aggregation varies between $0\% - 22\%$ which increases the power consumption at the sending node lineraly to the packet size.

Finally note that we can dramatically reduce the computation costs for addition at the aggregator nodes when shifting the division operations to the more powerful sink node. For $d = 2$, $d = 3$, and $d = 4$ the clock cycles for addition of ten operands at the aggregator node decrease to 80, 120, and 160 clock cycles which means that our approach is beating the computation effort of the competitor in any case. On the other hand this comes at the cost of additonal transmission effort since the size of the operands to be transmitted may increase in some cases. Another limitation is that this optimization has only value in WSN topologies with a single hierarchy level of aggregator nodes. Consequently, its main advantage is for aggregation functions which do not support Eq. 15. Here an aggregator node may be responsible for hundreds of sensor nodes. E.g, for adding 100 operands of size one byte with $d = 3$, 1200 clock cycles are required. On the contrary the hop-by-hop based encryption approach would require 23836 clock cycles.

10. CONCLUSION AND FUTURE WORK

We introduced the problem of end-to-end encrypted data aggregation in WSNs. We showed that privacy homomorphisms are encryption transformation with particular characteristics valuable for concealed data aggregation. By applying the privacy homomorphism proposed by Domingo-Ferrer we showed its suitability to aggregation functions average and movement detection. Actual implementation and its performance comparison with a hop-by-hop encryption scheme confirms that the approach is feasible and frequently even more energy saving than hop-by-hop encryption addressing a much weaker attacker model.

Currently we are investigating another PH. We term it conditional PH since the operands cannot be chosen fully independently of each other. Nevertheless their interdependencies still gives enough degree of freedom that the approach fits to the objectives of concealed data aggregation. The conditional PH allows addition, subtraction, multiplication and inverse multiplication with a skalar at minimal computation costs. The latter even guarantees a more flexible appliance to hierarchical WSN topologies under the assumption of an appropriate aggregation function.

11. ACKNOWLEDGEMENTS

The authors wish to thank Josep Domingo-Ferrer for his valuable feedback and co-operation.

This work is supported in part by the EU Framework Programme 6 for Research and Development (IST-2002-506997). Results are part of the DAIDALOS project (http://www.ist-daidalos.org).

12. REFERENCES

[1] R. Anderson, F. Bergadano, B. Crispo, J. Lee, C. Manifavas, R. Needham, "A new family of authentication protocols" In *ACM Operating Systems Review*, 32, 1998.

[2] M. Bohge, W. Trappe, "An Authentication Framework for Hierarchical Ad Hoc Sensor Networks" In *2nd ACM Workshop on Wireless Security (WiSe'03)*, pp. 79-87, September 2003.

[3] A. Boulis, S. Ganeriwal, M.B. Srivastava, "Aggregation in sensor networks: an energy-accuracy trade-off" In *Elsevier journal of Ad Hoc Networks*, Volume 1, Issues 2-3, pp. 317-331, September 2003.

[4] J. Deng, R. Han, S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks" In *1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, pp. 83-93, October 2003.

[5] M. Horton, D. Culler, K. Pister, Jason Hill, R. Szewczyk, A. Woo, "MICA, The Commercialization of Microsensor Motes" In *Sensors*, Vol. 19, No. 4, pp 40-48, April 2002.

[6] A. Perrig, R. Canneti, D. Song, J.D. Tygar, "The TESLA Broadcast Authentication Protocol" In *RSA Cryptobytes*, Summer 2002.

[7] A. Perrig, R. Szewczyk, V. Wen, D.E. Culler, J.D. Tygar, "SPINS: Security protocols for sensor networks" In *Mobile computing and Networking*, pp. 189-199, 2001.

[8] R.L. Rivest, L. Adleman, M.L. Dertouzos, "On data banks and privacy homomorphisms" In *Foundations of Secure Computation*, Academia Press, 1978, pp. 169-179.

[9] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism", In *Information Security Conference*, LNCS 2433, pp. 471-483, 2002.

[10] B. Przydatek, D. Song, A. Perrig, "SIA: Secure Data Aggregation in Sensor Networks", In *1st ACM Workshop on Sensor Systems (SenSys'03)*, November 2003.

[1] TinySec only supports the modes "No TinySec", "TinySec Authentication", and "TinySec Authentication and Encryption", which makes it difficult to solely measure the overhead for encryption.

[11] C. Karlof, N. Sastry, D. Wagner http://www.cs.berkeley.edu/ nks/tinysec/.

[12] D. Wagner, "Cryptanalysis of an Algebraic Privacy Homomorphism" (revised version), In *Proceedings of the 6th Information Security Conference (ISC03)*, Bristol, UK, October 2003.

[13] A. Weimerskirch, D. Westhoff, "Identity Certified Zero-Common Knowledge Authentication" In *1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, pp. 33-40, October 2003.