CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks

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

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

Markus Schneider
Fraunhofer SIT
64296 Darmstadt, Germany
E-mail: markus.schneider@sit.fhg.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. We present an approach that (1) conceals sensed data end-to-end, by (2) still providing efficient in-network data aggregation. The aggregating intermediate nodes are not required to operate on the sensed plaintext data. We apply a particular class of encryption transformation and exemplarily discuss the approach on the basis of two aggregation functions. We use actual implementation to show that the approach is feasible and flexible and frequently even more energy efficient than hop-by-hop encryption.**

I. Introduction

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. In particular we can not assume a sensor node to comprise a tamper-resistant unit. 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 and even on monitoring the health status of cattle stocks.

Analysis in most scenarios presumes computation of an optimum, e.g., the minimum or maximum, the computation of the average, or the detection of 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 the wireless connection. Since the energy consumption increases linearly with the amount of transmitted data, an aggregation approach helps increasing the WSN's overall lifetime. Another way to save energy is to only maintain a connected backbone for forwarding traffic, whereas nodes that perform no forwarding task persist in idle mode until they are re-activated.

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. A sensor node S_i, $i = 1, ..., n$ monitors environmental data s_i and sends them to an aggregator node which subsequently performs the aggregation function $y = f(s_1, \ldots, s_n)$ with $f : \{0,1\}^k \times \ldots \times \{0,1\}^k \rightarrow \{0,1\}^{k+l}$ on all incoming data. An aggregator node either transmits y to the sink node or to another aggregator node which again performs aggregation. This communication may even be multi-hop. Aggregator nodes belong to the backbone whereas sensor nodes persist in an idle mode until the sink node initiates a process which requirers a subset of them to contribute. To balance energy comsumption aggregator nodes should be periodically elected. The sink, which is assumed to be more powerful node, may either be the connection to the fixed network or the end point for the data collection process.

In this work, we consider WSNs in which messages should be transferred in a confidential way. It is our aim that passive adversaries that eavesdrop communication between the sensors, aggregators, and the sink, cannot obtain the exchanged information. This is achieved by encrypting transmitted data. Other security goals, such as integrity, are outside the scope of this paper. Furthermore, we assume that our class of adversaries can exclusively carry out ciphertext-only attacks. In principle, there are several possibilities in order to achieve the security goal. If end-to-end encryption is desired, then applying usual encryption algorithms, e.g., RC5 which is used in TinySec [11], implies that intermediate nodes have no possibility for efficient aggregation allowing to shrink the size of messages to be forwarded. The application of usual encryption algorithms combined with the requirement of efficient data aggregation provides only the possibility of encrypting the messages hop-by-hop. However, this means that an aggregator has to decrypt each received message, then aggregate the messages according to the corresponding aggregation function, and finally encrypt the aggregation result before forwarding it. Furthermore, hop-by-hop encryption possess that intermediate aggregators require keys for decryption and encryption.

The major contribution of this work is the provision of

end-to-end encryption for reverse multicast traffic between the sensors and the sink node. The proposed approach provides aggregators with the possibility to carry out aggregation functions that are applied to ciphertexts. This provides the advantage that intermediate aggregators do not have to carry out costly decryption and encryption operations, and thus, do not require to store sensitive cryptographic keys. The latter ensures an unrestricted aggregator node election process for each epoch during the WSN's lifetime which is impossible in case of hop-by-hop encryption. Here, only nodes which have stored the key can act as an aggregator node and thus, restricts the possibilities for energy balancing.

The rest of the paper is organized as follows. Section II provides some related work. Section III introduces a particular class of encryption transformations, namely privacy homomorphisms. In Section IV, we describe the reference privacy homomorphism proposed by *Domingo-Ferrer*. In Section V this approach is applied to the problem of concealed data aggregation in WSNs. Sections VI and VII give a proof of concept according to the specific aggregation functions *average* and *detect moving entity*. Parameter settings are discussed in Section VIII. In Section IX we discuss the concealed data aggregation approach for appliance in hierarchical WSN topologies before, in Section X, we show how it fits to the requirements of a particular destination platform. Finally, Section XI contains our conclusions and future directions.

II. Related Work

The focus of this work is on a solution for confidential data exchange in WSNs that supports data aggregation. To our best knowledge this is the first work proposing a solution for end-to-end encryption under such circumstances. The proposed solution assumes passive adversaries. In practice, there are several other security goals that should be fulfilled by combining other mechanisms, e.g., authentication of communicating sensors, protection of data integrity, and plausibility of sensed data. The proposals regarding other protection goals in WSNs especially focus on integrity and plausibility of sensed data. For the first, Bohge and Trappe in [1] provided on base of Perrig's TESLA [5] an authentication framework for hierarchical ad hoc sensor networks which to some extent may support our contribution. Further candidates for the authentication of sensor nodes and the integrity of sensed data may be μTESLA [6], ZCK [14], [16] or IC [15]. Approaches that deal with plausible data aggregation are from Boulis et al. [2] and SIA [10] from Przydatek et al. They focus on the efficiency-accuracy trade-off for computing plausible aggregation data.

III. Privacy Homomorphisms

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 $E : K \times Q \rightarrow R$ and the corresponding decryption transformation $D : K \times R \rightarrow Q$. 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. [7]. Generally, the more operands a PH supports the more computation intensive the transformations E and D are. For instance, RSA is a multiplicative PH. In [8] Domingo-Ferrer presented an additive and multiplicative PH which is a symmetric scheme and secure against chosen ciphertext attacks. In [13] Wagner showed that the proposed PH is unsecure against chosen plaintext attacks for some parameter settings. We argue that for the WSN data aggregation scenario the security level is still adequate and use this encryption transformation as a reference PH. However, asymmetric PHs like proposed by Okamoto and Uchyama [12] although providing security as secure as factoring, are not acceptable in the context of WSNs due to execution times twice as slow than eliptic curve cryptosystems. We argue that for an adversary that wants to obtain some confidential information, it is only reasonable to break a mechanism if the costs for breaking it are lower than the value of the revealed information. We assume that the typical information exchanged in a WSN is not of extremely high value for adversaries.

Remark: Clearly, encryption schemes like RC5, IDEA or RC4 provide a higher security level and consume much less execution times [9] like any currently available symmetric PH. Unfortunately, applied in WSNs these schemes run into a security/flexibility trade-off. With a single network wide key the process of aggregator node election remains as flexible as possible at the cost of almost no security. A single corrupted node would reveal the information of the whole network. With cluster-wide keys, the security level of the WSN increases at the cost of almost static routing pathes and a fix set of unbalanced aggregators in the backbone. This fact is based on the observation that in systems without any tamper-resistant unit the weakest security component is not the cryptoscheme itself but the storage policy of sensitive data.

IV. An Additive and Multiplicative PH

We describe the parameter settings, encryption transformation and decryption transformation of the PH proposed by Domingo-Ferrer. The PH is *probabilistic* which means that the encryption transformation involves some randomness that chooses the ciphertext corresponding to a given cleartext from a set of possible ciphertexts.

Settings: The public parameters are a positive integer $d \geq 2$ and a large integer g. It is important that g has 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 $r^{-1} \bmod g$ exists and $\log_{g'} g$ is a indication to the security level.

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 $a_1, \ldots, a_d$ such that $a = \sum_{j=1}^{d} a_j \bmod g'$ and $a_j \in \mathbb{Z}_{g'}$.

Compute

$$E_k(a) = (a_1 r \bmod g, a_2 r^2 \bmod g, \ldots, a_d r^d \bmod g). \quad (3)$$

Decryption: Compute the j-th coordinate by $r^{-j} \bmod g$ to retrieve $a_j \bmod g$. In order to obtain 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.

V. Encrypted Data Aggregation

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 $y' = f(s'_1, \ldots, s'_n)$. 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.

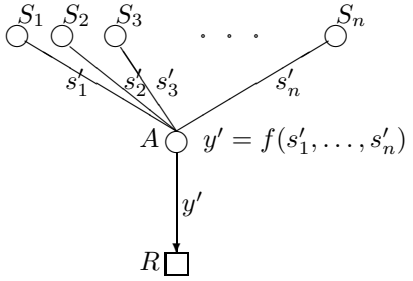

Figure 1: Concealed Data Aggregation for WSNs with PH.

More precisely, the concealed data aggregation for a WSN with the reference PH works as follows:

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

VI. 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

$$\begin{aligned} s'_1 &= E_{(3,7)}(1) = E_{(3,7)}(4,4) = (12,8) \\ s'_2 &= E_{(3,7)}(2) = E_{(3,7)}(7,2) = (21,18) \\ s'_3 &= E_{(3,7)}(1) = E_{(3,7)}(6,2) = (18,18) \\ s'_4 &= E_{(3,7)}(0) = E_{(3,7)}(3,4) = (9,8) \\ s'_5 &= E_{(3,7)}(1) = E_{(3,7)}(3,12) = (9,24) \end{aligned} \quad (5)$$

and transmit $S_1 \to A : (12, 8)$, $S_2 \to A : (21,18)$,$S_3 \to A : (18, 18)$, $S_4 \to A : (9, 8)$, and $S_5 \to A : (9, 24)$. A computes

$$\begin{aligned} y' &= \sum_{i=1}^{n} E_{(3,7)}(s'_i) \\ &= (12 + 21 + 18 + 9 + 9 \bmod 28, \\ &\quad 8 + 18 + 18 + 8 + 24 \bmod 28) \\ &= (13, 20) \end{aligned} \quad (6)$$

and transmits $A \to R : (13, 20)$. R computes

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

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

VII. 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*.

Again, we assume $n = 5$ sensors now monitoring movement patterns, say $(s_5, s_4, s_3, s_2, s_1) = (0, 0, 0, 1, 1)$, each in a perimeter with radius r. 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_i's 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 sensor network 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 node S_i which has monitored no movement sends the value $s_i = 0 \in \mathbb{Z}_{g'}$ to the A whereas in case of a movement detection, S_i sends $s_i = 2^{i-1} \in \mathbb{Z}_{g'}$ to A. Although the reference PH 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 fresh value $l \in \mathbb{Z}_{g'}$ to the sensors. A sensor S_i adds l to s_i and transmit the result to the A. Thus we extended the set of cleartext to $|Q| = 2g'$ at the cost of some pre-established additional group key for encryption between the S_is and the R. This we do although the PH from Domingo-Ferrer itself is probabilistic. Nevertheless our extension helps reducing the probability for a chosen plaintext attack since in addition it increases the plaintext space.

For a more detailed description let $g = 56$ and $(r, g') = (3, 14)$. Again, we choose $d = 2$. R chooses $l = 2$ and broadcasts its ciphertext concatenated with the aggregation function to the WSN. We exemplarily describe the computation at sensor nodes S_2 and S_3 for the sensoring tuple above: Since 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 $s_2 = 4$. Since S_3 has not observed any movement it computes $s_3 = 0 + l = 2$. Subsequently, the sensors apply Domingo-Ferrer's encryption transformation, i.e., sensors S_1 to S_5 compute:

$$\begin{aligned} s'_1 &= E_{(3,14)}(3) = E_{(3,14)}(2, 1) = (6, 9) \\ s'_2 &= E_{(3,14)}(4) = E_{(3,14)}(11, 7) = (33, 7) \\ s'_3 &= E_{(3,14)}(2) = E_{(3,14)}(2, 18) = (6, 28) \\ s'_4 &= E_{(3,14)}(2) = E_{(3,14)}(1, 1) = (3, 9) \\ s'_5 &= E_{(3,14)}(2) = E_{(3,14)}(13, 17) = (39, 41) \end{aligned} \quad (8)$$

and transmit the results to aggregator A. Then, A computes

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

before transmitting it to R. Subsequently, R computes

$$\begin{aligned} y &= D_{(3,14)}(y') \\ &= (31 \times 19 \bmod 56, 38 \times 19^2 \bmod 56) \bmod 14 \\ &= (29, 54) \bmod 14 \\ &= 13. \end{aligned} \quad (10)$$

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

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

The decrypted value is 3 and its binary representation in the 5-tuple is $(s_5, s_4, s_3, s_2, s_1) = (0, 0, 0, 1, 1)$. From this information R can infer that an entity has moved from (x_1, y_1, r) to (x_2, y_2, r) or vice versa.

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 let R be aware of n and m. Each $S_{(i,j)}$ now transmits $s'_{(i,j)}$ meaning that A receives $n \cdot m$ sensed and encrypted values.

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

$$y' = \sum_{i=1}^{n} \sum_{j=1}^{m} E_{(r,g')}(s'_{(i,j)}) \quad (11)$$

and transmits y' to R. Then, 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 separated n-tuples. Assume, R separated 3 tuples each of 5 elements

$$\begin{aligned} (s_{(1,1)}, s_{(1,2)}, s_{(1,3)}, s_{(1,4)}, s_{(1,5)}) &= (0, 0, 0, 1, 0) \\ (s_{(2,1)}, s_{(2,2)}, s_{(2,3)}, s_{(2,4)}, s_{(2,5)}) &= (0, 0, 1, 0, 0) \\ (s_{(3,1)}, s_{(3,2)}, s_{(3,3)}, s_{(3,4)}, s_{(3,5)}) &= (1, 1, 0, 0, 0) \end{aligned} \quad (12)$$

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.

VIII. Parameter Discussion

Applying the reference PH to a WSN 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 platform and its radio stack as we present in Section X.

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. For instance, for additive homomorphic operations, if the cleartext set counts $|Q| = 256$ elements and the information of $n = 10$ sensors are bundled by 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 the reference 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.

Table 1: Computation effort for CDA with reference PH.

	encrypt (at S)			add (at A)			decrypt (at R)		
d	$+$	$\times$	%	$+$	$\times$	%	$+$	$\times$	%
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 e.g. Crossbow's Mica Motes, operands larger than 8 bits have to be handled through the use of special software 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. For the measurements depicted in Table 1, we set the value of g not larger than 4 bytes.

Summing it up: Since 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 g should be used in a balanced way to ensure on the on hand an approppriate level of security and on the other hand only moderate computation. We feel that g in the range of 2^{32} is an appropriate choice for the envisioned scenario.

IX. Appliance to a Hierachical WSN

Next, we discuss how the concealed data aggregation can be used in a hierarchical manner. Independently of the underlying PH's algebraic properties a hierarchical concealed data aggregation only holds if the aggregation function itself has the following characteristic:

$$f(s_1, \ldots, s_n) = f\big(f(s_1, \ldots, s_i), f(s_{i+1}, \ldots, s_j), \ldots, f(s_{n-k}, \ldots, s_n)\big) \quad (13)$$

Unfortunately, the aggregation functions *movement detection* and *average* do not support this characteristic. However, the aggregation function *average* can be implemented by applying the *sum* function which conforms to Eq. (15). This means that aggregators perform the *sum* function instead of the *average* function. If we assume that the sink node knows the number of sensor nodes n that have sent their values to aggregators, then the sink node can easily divide the decrypted sum value by n. As a side effect with this approach only the sink needs to know n. Note that also in a hierarchical aggregator scenario, encryption is only done at the leaves (sensoring nodes). Decryption is exclusevly done at the powerful sink node.

X. Real World Consideration

In this Section we present measurements from our implementation in TOSSIM [4] and show how applying CDA helps
- distributing the overall energy consumption in a balanced way, and
- reducing 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 risk 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. 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. 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.

We evaluate the performance of our approach for the Mica2 Motes [3] with an Atmega 128 CPU 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, division (modular operation) and bit operations. Although they do not contemplate all processor instructions used in the algorithms' 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. We do not aim at an absolute value study for our implementation but rather a comparative study.

Table 2: CDA with Domingo-Ferrer PH vs. TinySec's RC5 execution times for a Mica2 Mote in clock cycles [cc].

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

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 the reference 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.

At a first glance the above measurements clearly indicate the reader's 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 is ben-

eficial 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 ($1452 \lessapprox (n+1) \cdot 236$). Assuming each aggregator node to be responsible for ten sensor nodes the reference PH still takes 1452 clock cycles whereas clock cycles for applying hop-by-hop encryption are nearly twice as much.[1] 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. 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 balance the energy consumption this disadvantage is acceptable since the aggregator node clearly is the performance bottleneck when maintaining a connected WSN backbone. To recall, contrary to aggregator nodes, sensor nodes persist a considerable period of their lifetime in idle mode.

When considering the radio stack, for the Mica Motes, a TinyOS (TOS) packet is pre-configured 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 [2] whereas the corresponding reference 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.

XI. 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 additive PH from Domingo-Ferrer as a reference PH our proof of concept indicates the principle suitability of symmetric additive PHs 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 for a broad range of realistic WSN topologies even more energy saving than hop-by-hop encryption. Especially if aggregator nodes need to be elected per epoch anew we advocate the usage of CDA. Future work will consider two aspects. Continuation of this work includes the support of other PHs in CDA and impact of the scheme on the flexibility of the network.

[1] We can dramatically reduce the computation costs at the aggregator nodes when shifting the division operations to the more powerful sink node. The clock cycles for addition of ten operands at the aggregator node decrease to 80, 120, and 160 clock cycles for varying d, which means that our approach is beating the competitor in any case. This comes at the costs that this optimization has only value in WSN topologies with a single hierarchy level of aggregator nodes.

[2] 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.

XII. 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).

References

[1] 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.

[2] 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.

[3] 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.

[4] P. Levis, N. Lee, M. Welsh, D. Cullar, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications", In *First ACM Conference on Embedded Networked Aensor Systems (SenSys) 2003, Nov. 2003*.

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

[6] 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.

[7] 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.

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

[9] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, M. Sichitiu, "Analyzing and Modeling Encryption Overhead for Sensor Network Nodes" In *ACM International Workshop on Wireless Sensor Networks and Applications, (WSNA'03)*, in conjunction with ACM MobiCom 2003, San Diego, Californa, USA, September 2003.

[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.

[11] C. Karlof, N. Sastry, D. Wagner "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks.", In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[12] T. Okamoto, S. Uchiyama, A new Public-Key Cryptosystem as Secure as Factoring, In *Advances in Cryptology - EUROCRYPT'98*, pp. 208-318, 1998.

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

[14] A. Weimerskirch, D. Westhoff, "Zero-Common Knowledge Authentication for Pervasive Networks", In *10th Selected Areas in Cryptography, SAC'03*, Springer-Verlag LNCS, pp. 73-87, August 2003, Ottawa, Ontario, CA.

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

[16] A. Weimerskirch, D. Westhoff, S. Lucks, E. Zenner, Efficient Pairwise Authentication Protocols for Sensor Networks: Theory and Performance Analysis, In *IEEE Press: Sensor Network Operations*, Editors: J. Carruth, T.F. La Porta, IEEE Press Monograph, September 2004.