# Analyzing Homomorphic Encryption Schemes in Securing Wireless Sensor Networks (WSN)

**Levent Ertaul[†], Johan Hadiwijaya Yang[†], and Gokay Saldamli[††]**

[†] Department of Mathematics & Computer Science, CSU East Bay, Hayward, CA, USA
[††] MIS Department, Bogazici University, Bebek, Istanbul, Turkey

**Summary**
Data aggregation in WSN (Wireless Sensor Networks) is substantial in increasing the network lifetime by eliminating the information redundancy. However, in current practice, aggregation data is transmitted in clear hence the whole process is prone to various attacks. In the past, this unwanted pitfall was mostly due lack of efficient encryption technology suited for limited WSN nodes; however, relatively lightweight HES (Homomorphic Encryption Schemes) allowing operations on the encrypted data is considered as a promising solution in securing such constrained devices. In this study, we carry out a comprehensive performance analysis of the most popular HES, particularly targeting limited WSN nodes. To our measurements: the implemented HES primitives match the performance and low power requirements; are feasible to be deployed inside the current widely deployed sensor nodes, and are scalable to thousands of sensor nodes without straining the lifetime of the whole network.

***Key words:***
*WSN, data aggregation, homomorphic encryption.*

## 1. Introduction

The development of tiny electronic devices has reached to a new era that it is possible to have mechanical elements, like cantilevers or membranes, to be manufactured at a scale more akin to microelectronics circuit than to lathe machining [1]. This technology, MEMS (Micro Electro Mechanical Systems), in turn is aiding the birth of WSN [2][3] consists of individual nodes that are able to interact with their environment by sensing or controlling physical parameters; these nodes have to collaborate in order to fulfill their tasks as, usually, a single node is incapable of doing so; and they use wireless communication to enable this collaboration.

The sensor node by itself has limited capabilities: limited computing resource (limited computing ability, small storage, and limited power source), unreliable type of communication (unreliable data transfer, limited data rate, and limited communication range) and unattended operation (limited trust, complex remote management). However, by adding more and more nodes, these seemingly limited devices can create a powerful infrastructure for a particular application sensing the wide area environment [2][3][4].

Due to their inherent limitations, WSN poses new challenges not present in traditional networks. Therefore, to develop useful mechanisms--- while borrowing the ideas from existing approaches--- it is necessary to identify and to understand those challenges. Regarding various possible critical applications, security is one of the most important aspects of the technology. The security needs to begin at the design stage because once we mass to deploy them it is impossible/infeasible to retrieve the nodes [5][6].

The two most pressing issues on securing WSN are node capture and power conservation. Node capture is most likely to happen for WSN since most of WSN nodes are deployed in an open area; hence are prone to attacks. Power conservation is critical to WSN lifetime. Since most of the nodes are battery-bound, once deployed, it is virtually impossible to replace/recharge the battery. Although other alternative energy source have been proposed [7], the developments are still in their infancy to be considered as a total battery replacement. Approaching the problem in a different manner, i.e. data aggregation [8] technique solves the power conservation issue; however, it does not provide data confidentiality. Therefore, using HES we intent to secure the data aggregation process which implicitly solves both of the mentioned pressing issues.

To be more specific; we implement four of the most well-suited HES [9][10][11][12] on WSN nodes and perform a comprehensive analysis focusing performance, feasibility and scalability. Performance is measured by how fast our algorithms perform on constrained WSN nodes which in turn determine the quality of service of the security realization. Feasibility is measured by contrasting several algorithms parameters to determine if the implementation is feasible; this knowledge aids the developer to make better a decision before the actual WSN deployment. Scalability of WSN implies the lifetime of WSN which is measured by how long the network can provide its service in the scale of thousands of nodes [13].

In the next two sections, we discuss the preliminaries. After giving main characteristics of WSN which are unique in the computing industry, we cover the main aspect of the

Manuscript received May 5, 2015
Manuscript revised May 20, 2015

targeted HES i.e. given in [9][10][11][12]. A brief description is followed by a known security related issues. In Sec. 4, we study the simulation aspects. After setting the basic requirements for simulation; we compare several simulator frameworks and choose OMNet++ as our main development platform. Using OMNet++, we formulate simulation goals, focus, and limitations. A steady state simulation is formed on three implementation levels: node, cluster, and network level. By profiling these three levels, we get a better understanding of power usage for each WSN device when performing the HES implementation. We further explore different alternative scenarios which show the effect of different parameters to test the solution's upper bounds. Knowing the upper bounds enables the WSN developers to make better decisions in deploying WSN in real world applications. In Sec. 5, we summarize the HES performance, feasibility and scalability on WSN. The optimal parameters in implementing HES on WSN are also presented. Last, we mention further research that needs to be done for making the implementation more feasible.

## 2. WSN security and data aggregation

In the nutshell, the wireless sensor network (WSN) is the combination of a CPU, a sensor, a radio, and is powered by a battery. Due to its rapid development from advancements in electronics, mechanics, computing and networking; especially of the breakthroughs in miniaturization technology called MEMS [1]. It is envisioned that WSN consisting of thousands to millions of tiny sensor nodes will change the way we obtain information from the physical environment. A node by itself is rather limited; however, when networked together, these devices can provide high resolution information about sensed phenomena. Possible applications of WSN range from natural habitat sensing, to structural monitoring, to emergency response, and to military application [3][4] [14][15]. Reasonably, there has been a great surge of interest in WSN focused on developing hardware, software and networking architecture needed to enable such applications.

Like any other constrained environment, WSN poses new challenges not present in traditional networks. To be more informative; Fig. 1 compares WSN with other technologies, from the most to least restricted, in terms of computing power and price per unit. Observe that, RFID (Radio Frequency Identification) [16] having only the object identification functionality has both the lowest computing power and price per unit. Next token like devices are smart cards, a step up from RFID with the additional capability to process the data. WSN devices are between smart cards and MANET (Mobile Ad-hoc Network) technology [5].

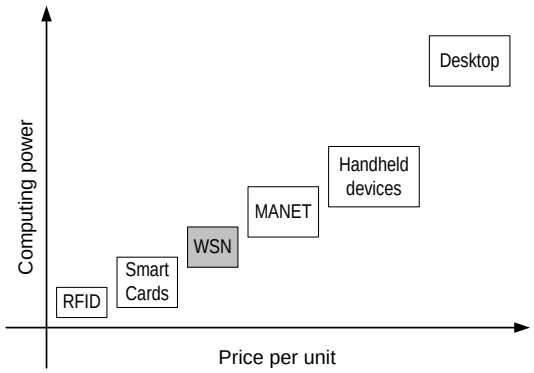


Fig. 1. WSN compared with other technologies.

Although WSN node could be considered cheap in terms of price per node, the information in each node might be many more times more valuable. Traditionally, security is considered as an add-on to existing arrangements. However, having such a huge variety of potential applications, WSN needs to have security in place. This is due to the fact that in many application scenarios, once a node is deployed, it is virtually impossible to change the settings afterwards [14]. Therefore, WSN security solution needs to be implemented during design development

Because of being a of a data driven network, WSN raises new threats that are different from what we have faced before. WSN allows massive data collection, coordinated analysis, and automated event correlation. For instance, consider a sensor network used for tracking people and vehicles over long periods of time, with troubling implications [13]. Facing such implications, WSN is required to have security in place to ensure confidentiality, authenticity, integrity, availability, reliability, and scalability. In providing the above security requirements, there has been a great effort in security community. Among these studies, some of the key solutions could be pointed as follows: cryptography [5][6][17]; key management [18][19]; authentication [20]; secure routing [21]; location aware security (Key establishment [22], privacy aware [23], location verification [24]) and data aggregation [8][25][26][27] [28][29][30][31][32].

In this study, we focus on data aggregation using HES primitives. In a network of thousands of sensor nodes, the data from an individual node is not meaningful when compared to the aggregated data from clusters of nodes. In this case, the raw data from several nodes will be buffered and aggregated in one node acting as aggregator before

sending a single message representing the aggregate of values to nodes upstream. Data aggregation substantially cuts down the transmission costs and in turn keeps the network available for a longer time and provides optimal bandwidth usage [8][29].

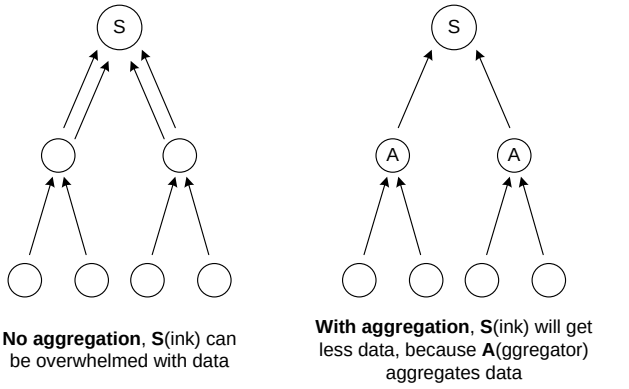


Fig. 2. Comparing the network traffic with and without data aggregation.

In Fig. 2, the effect of data aggregation to WSN traffic is illustrated. On the left, WSN operation without data aggregation is seen. Observe that since the middle nodes are not aggregating the data but just relaying the messages, as the network tree level gets deeper, the sink node gets more overwhelmed. On the other hand, shown on the right, the middle nodes acting as an aggregator summarize the data from lower nodes and hence the sink has lesser data to process [8][33].

Data aggregation solves the power conservation issue. However, in its implementation, the information flow through the network is mostly insecure. On top of that, any implemented security solution is added to the power consumption overhead, quickly depletes the sensor node power supply, and hence defeats the purpose of power conservation.

Since HES allows operation on ciphertext as if it was done on plaintext, it is an ingenious method of solving WSN data (aggregation) confidentiality. Being different from the other cryptographic security measures, HES requires only light computing demand and only use small amount of memory.

## 3. Homomorphic encryption schemes

Homomorphic Encryption Schemes (HES) offers significant advantages in securing WSN data aggregation, such as: low computational demand, long network lifetime, allowing distributed computing using untrusted nodes, not revealing sensitive information, and end-to-end security. Conventional cryptography can not solve the security problems due to the limitation of WSN. The HES has a great potential to solve data confidentiality requirement, because it allows computation on encrypted data as if on the plaintext data. If HES is not used to secure data aggregation, the intermediate nodes need to have the knowledge of secret keys to perform the decryption on the data before being able to do operation on them, and to re-encrypt the data; which are prone to attacks.

From the five types of HES operations: additive, subtractive, multiplicative, inverse multiplicative and mixed multiplicative; we found four HES [9][10][11][12] suitable to be implemented on WSN. In terms of computing demand, additive and subtractive homomorphism are lighter than the rest. In this study, we only apply the additive homomorphism (decrypting the sum of two ciphertexts is the same as addition of two plaintexts; $E(x+y) = E(x) + E(y)$ because it performs well on WSN and provides sufficient security for WSN data aggregation.

Next, we discuss Domingo-Ferrer schemes with the explanation of encryption and decryption process, the discussion of its security level, and the simple example that illustrate how the algorithm works.

### 3.1 Domingo-Ferrer a new privacy homomorphism

Domingo-Ferrer a new privacy homomorphism (DFPH) [9] is a HES which operates by splitting the message and encrypting the splits. The splits can be added or subtracted on the way to the destination. On the destination, the message is then decrypted. This way the data is concealed from source to destination. Let $d$ and $m$ be the public parameters of the scheme where $d$ is the number of plaintext splits and $m = pq$ for some secret large primes $p$ and $q$. In addition to $p$ and $q$, the scheme has two more secret parameters $x_p$ in $Z_p$ and $x_q$ in $Z_p$. Even though, the modulus $m$ is assigned as the public parameter, it could also be kept secret to increase the security. Alg. 1 gives the basics of encryption/decryption functions.

**Algorithm 1**: DFPH

**Encryption**

1. Choose $a$ in $Z_m$, such that $a < \min(p, q)$, (observe that $a = a \bmod m = a \bmod p = a \bmod q$)
2. Split $a$ into secret numbers $a_1, a_2, \ldots, a_d$; such that $a = \sum_{j=1}^{n} a_j \bmod m$
3. Compute $E_k() = ([a_1 x_p \bmod p, a_1 x_q \bmod q], [a_2 x_p^2 \bmod p, a_2 x_q^2 \bmod q], \ldots, [a_n x_p^n \bmod p, a_n x_q^n \bmod q])$

**Decryption**

1. Compute scalar product of the $j$-th pair [mod $p$, mod $q$] by $[x^{-j}_p \bmod p, x^{-j}_q \bmod q]$
2. add them up to get $[a_j \bmod p, a_j \bmod q]$;
3. use CRT (Chinese remainder theorem) to get $a \bmod m$

Note that this scheme is known to be secure against to known-ciphertext attacks, but not to the known-plaintext attacks [34].

### 3.2 Domingo-Ferrer allowing field operations on encrypted data (DFFO)

DFFO [10] generates secret random numbers and hide those numbers from the intermediate nodes. This scheme conceals the data from source to destination like as the previous DF method. Alg. 2 shows how the scheme works. Note that the public parameter $m = pq$ ($p$ and $q$ are large primes) is available for every sensor node where the secret parameters $p$ and $q$ are only available for the sensor nodes at the boundary of the network and the sink node.

**Algorithm 2**: DFFO

**Encryption**

1. $Q_p$ is defined as $Q_p = \{a/b \mid a, b \text{ in } Z_p\}$
2. Select a value $x$ in $Z_p$, a random fraction $a/b$ in $Q_p$, such that $x = ab^{-1} \bmod p$
3. The ciphertext is computed as $y = E_p(x) = ab^{-1} \bmod m$

**Decryption**

1. Pick any fraction $A/B$ in $Q_p$, such that $y = AB^{-1} \bmod m$
2. compute plaintext $x = D_p(y) = AB^{-1} \bmod p$ using $p$.

This scheme is secure against to chosen-ciphertext attacks, but not to known-plaintext attacks [35]. In other words; if $(x, y)$ is a known plaintext-ciphertext pair, finding key $p$ is relatively easy. The cryptanalyst can determine a set of $A_iB_{i-1}$ such that $AB^{-1} = y \bmod m$. We know $x = D_p(y) = AB^{-1} \bmod p$, where $p$ is the prime key and $x \leq p$. We also know that, $p \mid (A - xB)$, that is $p$ divides $A - xB$. Here $x$, $A$ and $B$ -1 are known, so finding $p$ is relatively easy.

### 3.3 Domingo-Ferrer additive and multiplicative privacy homomorphism (DFAM)

Likewise DFPH, DFAM [11] also operates on the splits of the message but the public and secret parameters are slightly different. While the public parameter $d$ still represents the number of plaintext splits, modulus $m$ should not be a product of large primes. In fact $m$ should have small divisors and many integers less than $m$ have to be invertible modulo $m$. On the other hand, there are three private parameters: $m'$ (an integer where $m/m'$ has 0 remainder), $r$ and $r_{inv}$ where $m' < r$ in $Z_m$ such that $r\, r_{inv} \bmod m = 1$. The detail of DFAM is given in Alg. 3.

**Algorithm 3**: DFAM

**Encryption**

1. Split $a$ into secret numbers $a_1, a_2, \ldots, a_d$ ; such that
   $a = (a_1 + a_2 + a_3 + \ldots + a_d) \bmod m'$
2. $E(a) = (a_1 r \bmod m, a_2 r^2 \bmod m, \ldots, a_d r^d \bmod m$

**Decryption**

1. Compute scalar product of the j-th pair $r^{-j} \bmod m$ to retrieve $a_j \bmod m$

This scheme is secure against to chosen-ciphertext attacks, but not to chosen-plaintext attacks. Wagner [35] showed an efficient way to recover $m'$ with a small pool say $n$, of known plaintexts. He proposed several ways to recover $r'$, where $r' \equiv r \bmod m'$. One possibility is exhaustive search, that works whenever $m'$ is small. Another possible attack is based on linear algebra, which works with reasonable success probability whenever $n \geq d$. A third possibility is an attack based on polynomial root-finding, which applies $m'$ can be factored and there are a few known plaintexts.

Cheon et. al [34] proposed a plaintext attack, such that if the attacker can guess several plaintexts, the key can be broken in polynomial time. The plaintext guessing in the context of WSN is easy to do, since the data collected on the nodes can yield the plaintext already. For example, the attacker can just measure the temperature and use it as the plaintext.

This fact should not hinder the usefulness of the scheme. Implemented properly, Ferrer's scheme is enough to increase the effort for the intruder to a degree that makes the attack uninteresting. The drawback of this approach is the size of the data is tremendously increased by the number of summands, the upper bound for range of numbers to be encrypted is $g'$, but the actual components of the encrypted values can be as large as $g$-1 which is waste of bits since $g'$ divides $g$ without remainder.

There is no multiplicative inverse makes it we have to take care not to exceed the limit $g'$. After reaching the limit $g'$, the problem called wrap around would happen which decreases the value by $g'$. The only solution if to decrypt a value, do the division and encrypt it again. The wrap around problem exists on additions though additions are less prone to exceed the limit. The non deterministic nature of Ferrer's scheme, such that the same number can map to many different ciphered words, provides a solid advantage over the intruders [35].

### 3.4 Mixed Multiplicative Homomorphism (MMH)

MMH [12] is similar to DFFO, it generates random numbers and it security depends on hiding those numbers from the intermediate nodes. Moreover, modulus $m$ is public and its large factors $p$ and $q$ are secret parameters. Alg. 4. gives the details of the encryption and decryption functions.

**Algorithm 4**: MMH

**Encryption**

1. $Q_p$ is defined as $Q_p = \{a/b \mid a, b \text{ in } Z_p\}$

2. Given $x$ in $Z_p$, pick a random number $a$ in $Q_p$ such that $x = a \bmod p$
3. The ciphertext is computed as $E_p(x) = a \bmod m$.

**Decryption**

1. Given $y = E_p(x)$ in $Z_m$. Use $p$ to recover $x$. i.e. $x = D_p(y) = y \bmod p$.

This scheme is secure against to known-ciphertext attacks, but not to known-plaintext attacks [12]. Let us look into the known-plaintext and known-ciphertext attacks in more detail regarding this cryptosystem.

1. Known plaintext attacks: If ($x$, $y$) are the known plaintext ciphertext pair, finding key $p$ is relatively easy. We know $x = D_p(y) = y \bmod p$, where $p$ is the prime key and $p \geq x$. We know that $p \mid (y - x)$, that is $p$ completely divides $y - x$. Here $x$ and $y$ are known, so finding $p$ is relatively easy as $p$ is the divisor of ($y - x$).
2. Known ciphertext attacks: We know $x = y \bmod p$ and if $y$ is known, $x$ and $p$ are still unknown. We know, $y = x + rp$ and it is difficult to determine $y$ as $x$ and $p$ are unknown.

Table 1 gives the summary of the four chosen DF algorithms. Observe that all of the algorithms can perform the additive homomorphism which in fact is the basis of our work. Additive homomorphism needs relatively lightweight computations and all of today's chips have the addition operator as their primitive function. Although multiplicative homomorphism might offer higher security, it requires more space and produces longer encrypted messages; hence it is not favorable for WSN use.

Table 1: Overview of the HES Algorithms.

| | Services | Against Secure | Against Unsecure |
|---|---|---|---|
| DFPH | $S(d_1,d_2) = d_1 + d_2$<br>$S(d_1,d_2) = d_1 \times d$ | KCP | KPA |
| DFFO | $S(d_1,d_2) = d_1 \pm d_2$<br>$S(d_1,d_2) = d_1 \times d_2$<br>$S(d_1,d_2) = d_1 / d_2$ | CCA | KPA |
| DFAM | $S(d_1,d_2) = d_1 \pm d_2$<br>$S(d_1,d_2) = d_1 \times d_2$ | CCA | CPA |
| MMH | $S(d_1,d_2) = d_1 + d_2$<br>$S(d_1,d_2) = d_1 \times d_2$ | CCA | KPA |

Note that schemes other than DFAM are insecure against KPA. The attacker does not need to feed the data into the captured device, only to know what kind of data is being fed. This is a serious weakness for WSN application which sense natural data, since the attackers can easily fabricated the value. However, this type of attack can be minimized in WSN because it is widely deployed and in order for attackers to succeed, they need to capture the majority of the nodes.

## 4. Simulation

In order to see the action of HES for data aggregation on WSNs, we need to model the real world implementation of sensor networks as approximately as possible. We simply manifest our simulation goals, focus, limitations, tools and possible scenarios in this section.

### 4.1 The simulation goals

The primary focus of this study is to analyze the performance, feasibility and scalability of four of HES [9][10][11][12] implementations on WSN. We compare the performance and the lifetime of the sensor nodes on the basis of how much the power usage and the execution time that a sensor node used in performing the security algorithm implementations. The CPU, memory, and the radio usage are the components of the measurements on power usage and execution time.

Feasibility is the other phenomena that we drill through the simulation by having a better understanding of the impact of different security algorithms parameters to the overall network lifetime. By looking at the simulation results, designers would be able to make decisions on their hypothesis. For instance, the implementation might use small CPU resource; but if it requires several transmissions to send because of some longer message; it will quickly drain power hence infeasible.

Having a large number of nodes heavily impacts simulation performance and scalability. Second, credible results demand an accurate characterization of the sensor radio channel. There is an increasing concern about the simulation methodology and assumptions used in simulation of WSN. Idealized hardware, simplified protocols, and unrealistic radio models too often lead to mistaken results.

### 4.2 The simulation focus

This is a steady state simulation which focuses in measuring the long term average behavior of WSN when executing the security algorithms. We do not focus on the behavior of WSN nodes at the starting point and assume the sensor nodes were already in a ready state, with network topology already defined, and all the bootstrapping processes done. While it is true that the bootstrapping process consumes energy, it is only done sparingly and will not have an impact on WSN long term performance. On the other hand, our security implementations are done repeatedly and hence greatly impact the network lifetime [7][36].

For WSN communication we focus on the reverse multicast communication. Most of WSN application is of the many to one communication pattern where many

sensors at the edge of perimeter will report back to one root point. The other types of communication patterns are temporal and therefore negligible for the long run [8][25].

### 4.3 Tools and framework

To understand the dynamic behavior of WSN mass deployment (in the order of thousands of nodes), we devise a robust simulation model based on an open source framework called OMNet++ (Objective Modular Network in C++). After researching several simulation options, OMNet++ fulfilled our basic requirements because of its extensibility, scalability, true imitation of real hardware and software and being easy to analyze and install [37].

We develop the WSN simulation on an Intel Pentium Xeon dual-core 2GHz, 2GB RAM and running Ubuntu Linux 9.10. Due to the in depth research and the wide adoption on current WSN applications, we based the simulation on MICA2 sensor nodes [38]. The success and the validity of the simulation relied heavily on the many additional contributing packages including: mobility framework (MF), Castalia, INET, and NesCT [37][38].

### 4.3 Simulation limitations

We do not focus on secure routing [21], key management [18][19] for authentication, dynamic aggregator node election, and the environmental effects. Even though all of those processes are important for WSN applications, they are temporal events hence negligible in impacting the network lifetime in the long run [8][25]. Instead, we use ready-made solutions provided by other research or from default settings provided by OMNet++.

### 4.3 WSN Simulation Design

We discuss the events that happened at sensor nodes and aggregator nodes. However, we do not focus on the events that happened at sink node because it is irrelevant to the long term performance of WSN.

The simulation process is a combination of event and unit time advance. Event advance focuses on a general view of the state changes each time an event occurs. On the other hand, when the simulation needs to capture more details of events that happened at the same time or close to each other; it changes to unit time advance. The simulation alternates between event advance and unit time advance, where event advance is active the time when nodes spend most of their time in the IDLE/SLEEP state and unit time advance actives during the time when nodes are in the ACTIVE state [38][39]. To keep track of the events, the simulation uses a data structure called future event list (FEL) to track two things: the time of occurrence of an event and the type of event.

The event advance of sensor node FEL is responsible to trigger an event which simulates a sensor node capturing the environmental value and to trigger an aggregator node to get ready to perform its task. On the other hand, the unit time advance captures more details of the sensor node activities by changing the simulation time from *ms* to $\mu s$. It captures the CPU, memory, and radio activities of the sensor node. After completing its tasks, the sensor node goes to IDLE/SLEEP state and the event advance takes over.

On the aggregator node, a similar alternating process goes on. There are two states of the aggregator node which will greatly determine its lifetime. The first is LISTEN state when the aggregator node spends most of its time waiting for sensor nodes to complete their data transmissions. The second is the IDLE state, when the aggregator node cannot get into SLEEP state because it has to relay messages from the aggregator nodes located below of the network tree. Since its activity depends on the completion of the other nodes, the longer it has to wait, the more it consumes the power source

### 4.3 Profiling

To measure the performance of the sensor nodes while performing HES algorithms, we profile them on two measurements: the execution time and the power usage. Since the measurement components are simply the sensor node's CPU, memory usage and radio, the simulation reveals the dynamics of execution time and energy usage within a node, a cluster, and a network.

We perform the profiling on the node, cluster, and network level. The **node level profiling** gives the power consumption details of the sensor node devices. Throughout this research, we model the sensor nodes based on MICA2 [38] specifications. According to this model, there are five WSN major activities: TX-transmitting, RX-receiving, COMPUTE, IDLE, and SLEEP. In case of TX-transmitting there are four devices on WSN sensor node that are active: the CPU is ACTIVE; the memory is ACTIVE; the radio is transmitting; and the sensor board is ON.

In Table 2, the snapshot of the consumed energy of WSN device per activity is given. For instance, on COMPUTE, the CPU consumed 8.93 *m*A, the memory consumed 12.3 *m*A, the radio consumed 3.7 *m*A, and the sensor board consumed 1.7 *m*A, for a total of 26.63 *m*A. After adding the subtotals of energy consumed per WSN activity, the grand total is 101.53 *m*A.

Once the sensor nodes are being deployed into one area, they begin to form a cluster which consists of several sensor nodes and one aggregator node. The sensor nodes are usually located at the edge where aggregator node is located in the middle of the network tree (see Fig. 3).

Table 2: Total amount of energy used per WSN device per activity (*mA*).

| Devices | TX | RX | Compute | Idle | Sleep |
|---|---|---|---|---|---|
| CPU | 8.93 | 8.93 | 8.93 | 1.04 | 4.13 |
| Memory | 12.3 | 12.3 | 12.3 | 0 | 0 |
| Radio | 10.07 | 7 | 3.7 | 3.7 | 3.7 |
| Sensor | 0.7 | 0.7 | 1.7 | 0.7 | 0.7 |
| Subtotal | 32 | 28.9 | 26.63 | 5.44 | 8.53 |
| Total: | 101.53 *m*A (100%) | | | | |

Aggregator node's main function is to aggregate the information from children nodes and to do operation on the information before sending the aggregated value upstream towards the sink/base station node.

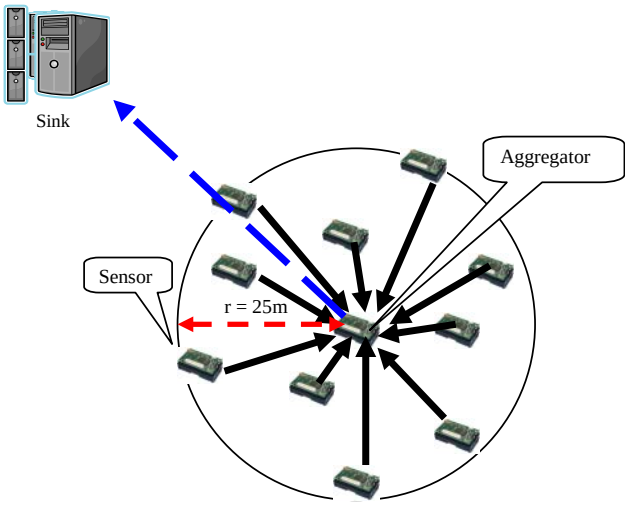


Fig. 3. Cluster level simulation configuration.

**Cluster level profiling** aims to reveal the interaction dynamics between the sensor and aggregator nodes within one cluster. Even though both devices consist of the same hardware specifications, each performs different functionality. For instance; the sensor nodes perform the encryption algorithms where the aggregator nodes, in our model, perform the operations (add, subtract, etc) on the encrypted values. The following gives the basic assumptions that we carry during our cluster profiling.

a) The sensor nodes are randomly distributed over an area with maximum radius of 25 meters based on an effective inter-nodes communication distant [36].
b) The aggregator node is located in the middle of the cluster and all the sensor nodes are reporting their data to the aggregator node acting as the cluster head.
c) The cluster uses a star shape network topology to share a common wireless radio communication channel.
d) After the event that triggers the sensor nodes, the data is being pushed to the aggregator node once in every sampling time interval. To see the effects to the performance and the lifetime of the sensor nodes, three different; i.e. 1, 2 and 4 minutes time intervals are used.
e) After the aggregator node polled the data, it sends the aggregated data to the sink node. The sink decrypts the data to reveal the actual message.

Lastly, the **network level profiling** gives the saturation details of the different sensor nodes to the performance and lifetime of the aggregator node. The network level FEL where many sensor nodes are simultaneously monitored, is the most complicated one to simulate.

The following procedure gives our simulation setup. Notice that the focus is more on performance of the aggregator nodes since they determine the lifetime of the network.

a) The network level simulation consists of clusters of nodes and one sink node.
b) The clusters are randomly distributed over (500×500) $m^2$ area.
c) The sink node is located in the middle of network.
d) Each cluster is headed by the aggregator node. The focus is on the aggregator node performance.
e) The network uses the fixed Star shape topology to share the wireless radio communication channels.
f) the sampling is carried in 1, 2 and 4 minutes time intervals where we see the effects to the lifetime of the aggregator node.

In our model, the simulation is built upon the previous cluster level profiling where we put clusters of nodes within one sensing area. In our measurements, we run-though two alternative scenarios; a network in which we gradually increase the number of clusters having fixed sensor nodes and a network in which we gradually increased the number of sensors within its fixed number of clusters.

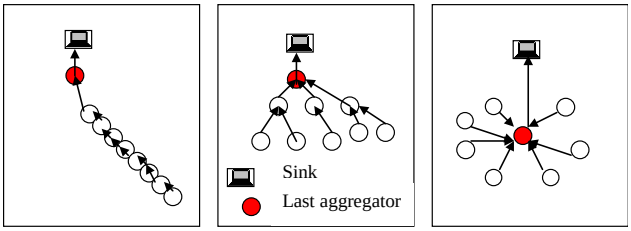


Fig. 4. Straight line (left), Balance Tree (middle), and Concentrated (right).

Another important concern is profiling the last aggregator node since it is directly connected to the sink. The last aggregator node --sometimes also called as the traffic concentrator--- drives the upper bound for the aggregator node's power usage and lifetime. There are three cases on profiling the last aggregator node as seen in Fig. 4:

i. **Straight line** -- the clusters form a straight line: the last aggregator node has to IDLE for long time because it has to wait all values from the edge of the network to reach the last aggregator node.
ii. **Balanced tree** -- the clusters are spread evenly to make a balanced tree: the last aggregator node has a shorter IDLE time.
iii. **Concentrated** -- the clusters are spread and each aggregator concentrates its data to the last aggregator.

Observe that the network of clusters forming a straight line gives the worst case scenario since the last aggregator node has to wait for all of the messages to reach it. In other words, the longer the aggregator node is active, the more of the power consumed. Thus, this would severely decrease the lifetime of the aggregator node. In the next section, we present the simulation results.

## 4.7 Simulation results

We start with the sensor profiling in performing the HES implementation on different key sizes, message sizes, and message splits. Our findings showing the node lifetime of ten encryption alternatives on comparable 512/512 bits (msg/key) parameters are presented in Fig. 5.

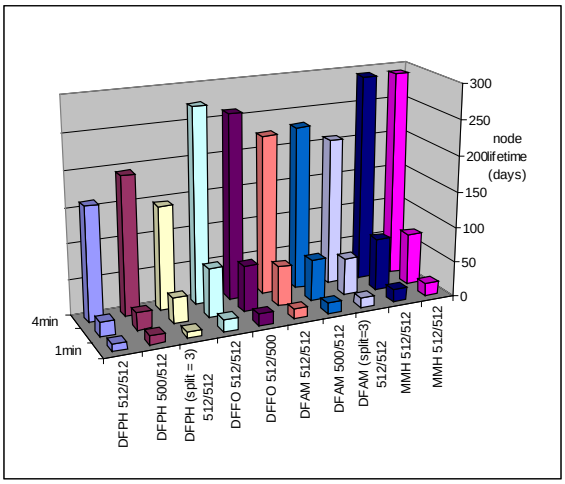


Fig. 5 Comparing lifetime of sensor nodes performing HES.

To be more informative, *y*-axis (in Fig. 5) shows the number of days the sensor survives; where *x*-axis gives the encryption schemes and *z*-axis shows three different sampling intervals: i.e. 1, 2, and 4 minutes sampling time intervals. Notice from the figure that MMH has the longest lifetime, followed by DFFO, DFAM, and DFPH. Both of MMH and DFFO use random values as secret keys and do not split the message; hence, they have a less computing overhead.

Note that DFPH on average has a shorter lifetime; particularly, whenever the message split is 3. The overhead of splitting the message is the need for bigger data structures to store and operate on the data splits. Observe that there is a significant increase in the lifetime when the sampling interval is increased. This is directly proportional to the amount of time a sensor is in SLEEP state which in fact, determine the lifetime of the node.

A reasonable comparison would be keeping the ratio of key and message size constant for all HES implementations. Fig. 6 depicts the lifetime of a sensor on 4 minutes of sampling interval in performing different HES implementations with the same key to message size ratio (128/128bits, 256/256 bits, 512/512 bits, etc). Observe that in such a setting, MMH and DFFO show no/slight change in the node lifetime regardless the size increase of both key and message. Nevertheless, both implementations serve better solution for their fast performance and longer lifetime. Meanwhile, DFPH gives a graph of a step function; the same lifetime for 128/128 and 256/256, and the same lifetime for 512/512 and 1000/1024 (msg/key). On the other hand, DFAM shows the rounder curve graph; which show that the lifetime of sensor node correlates with the message/key size.

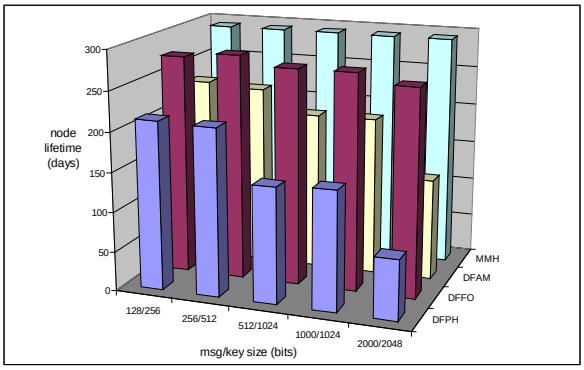


Fig. 6. Comparing lifetime of sensor nodes for HES deployments having the key and message size ratio equal to one.

To be more specific, we compare the node lifetime of two encryption schemes which use the message splits, DFPH and DFAM, in order to effect of different message splits to node lifetime.

In Fig. 7, we give three different msg/key ratios: 512/512, 512/1024 and 512/2048 on sensor node performing 4 minutes sampling time interval. The best lifetime is MMH, followed by DFFO. The worst lifetime is DFPH with message split 3 for msg/key: 512/2048. The effect of different message split to the node lifetime is substantial when the msg/key ratio is great; for instance, if we compare DFAM 512/512 split 2 and split 3 with DFAM 512/2048 split 2 and split 3; there is a substantial % difference for the latter scheme. We

postulate more message split require bigger data structure and longer ACTIVE CPU.

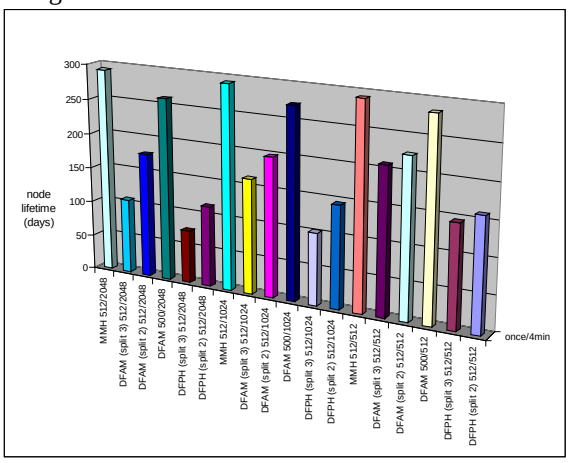


Fig. 7. Comparing node lifetime on 512/512, 512/1024, and 512/2048 (msg/key).

Next, we show the performance of an aggregator node in processing the aggregated data from its sensor nodes. These readings are important to formulate the baseline performance of one cluster. We show that the optimal solution is to increase the sampling interval time. The next solution is to decrease the number of sensors within one cluster; however, this solution might be infeasible for WSN applications that need higher sensing ability from higher node saturation per cluster. We observe from Fig. 8 that decreasing number of sensors within a cluster by half increases lifetime by average 70%. Decreasing number of sensors within a cluster by quarter increases lifetime by 181.95%. Increasing sampling time interval by factor of two (example from 1, 2, and 4 minutes sampling time intervals) increases lifetime by 300%.

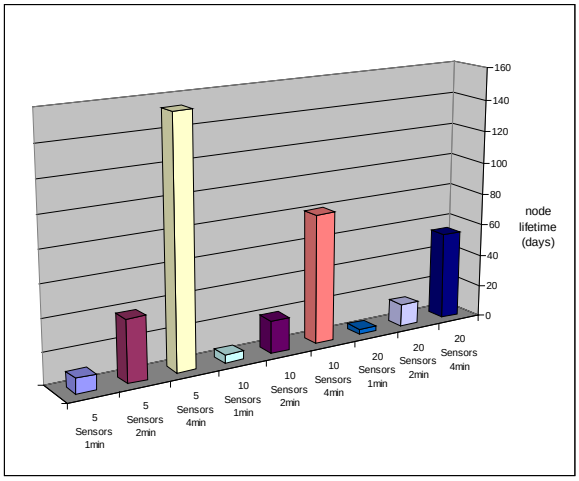


Fig. 8 Comparing aggregator lifetime on different number of sensors in a cluster and different sampling interval.

In order to see the effect of different number of clusters within one network to the power usage and the lifetime of the aggregator node. Since the last aggregator node drives the upper bound for the aggregator node's power usage and lifetime, we simulate the last aggregator node under the topologies seen in Fig. \ref{sec5_1:fig0}. The simulation parameters are: (i) All the nodes are performing DFPH with message split 2, 512/512 (msg/key), the sensing area is 500×500 square meters. (ii) Each cluster consists of 10 sensor nodes and one aggregator node, sampling time interval is 4 minutes. This sampling time interval is chosen to make an easier performance comparison, since the other sampling time interval results are too close to compare.

We give the result of this simulation focusing the last aggregator node in Fig. 9. On concentrated case, the last Aggregator spends most its time IDLE waiting for all data from the other aggregators to reach it. On average, adding more clusters to the network will reduce the Aggregator lifetime by 87.42% on worst case, as opposed to only 39.08% on best case scenario.

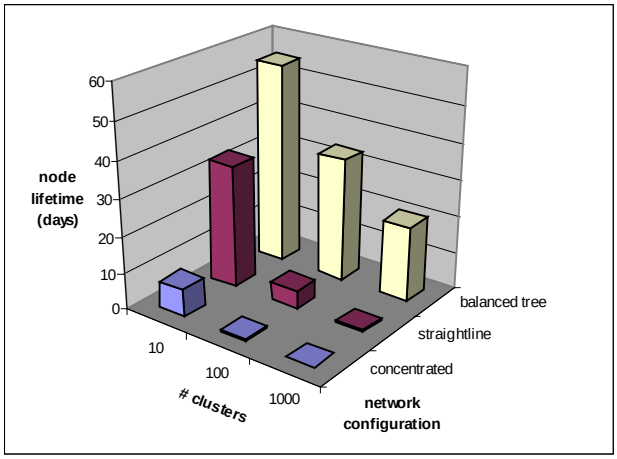


Fig. 9 Comparing different number of clusters within a Network to the last Aggregator node lifetime.

From the over all simulation results, it is clear that MMH is the fastest HES on the sensor node in comparison with the other three schemes. However, MMH is susceptible to known-plaintext attack. DFPH and DFAM share the worst performance (in terms of speed) because they split the value before encrypting each split. This requires more time to execute, more memory to process the bigger data, and consequently more power consumption. However, both schemes might provide a higher security level than MMH and DFFO because of the same reason that make them disadvantaged.

On the aggregator node, regardless of the schemes used, its performance depends on the number of sensor

nodes within its cluster and the network configuration. The more the sensor nodes are in the cluster, the more RX-Receiving operations the aggregator node is required to carry. The more unbalanced the network (not distributed evenly), the more time the aggregator node is required to wait for others to finish their operations. Both conditions reduce the aggregator node lifetime.

## 4. Conclusion

In this study, we have proven that all of the chosen HES algorithms can be implemented on and a good candidate in securing WSN's data aggregation despite of WSN's limitations. These algorithms do not require intensive calculations like in other conventional symmetric encryptions and public key cryptography. The maximum and minimum calculations can be done by randomize the pre-encrypted values; eliminate the encryption needs in sensor nodes. This will further prolong the lifetime of the sensor nodes. By prolonging the sensor nodes, in turn will prolong the whole network lifetime. We also have shown the direct correlations between the power usage and the execution time. The CPU, memory, and radio as the components of the power usage and the execution time; play the big role in determining the lifetime of the sensor node. The faster the execution of the algorithm, the longer the lifetime of the node for it can preserve much of its energy in the SLEEP state.

**Acknowledgments**

G. Saldamli is partially funded by TUBITAK research project 109E180.

## References

[1] F. Chollet and H. Liu, "A (not so) short intro- duction to MEMS," Nanyang Technology Univer- sity,MicroMachines Center, School of MAE, Singa- pore, Tech. Rep. 2.5, 2008.

[2] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in wireless sensor networks: theory and practical solutions," *Wireless Networks*, vol. 8, pp. 443–454, September 2002.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 88–97.

[4] K. Romer and F. Mattern, "The design space of wire- less sensor networks," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 11, pp. 54–61, 2004.

[5] S. Schmidt, H. Krahn, S. Fischer, and D. Watjen, "A security architecture for mobile wireless sensor networks." in *ESAS' 04*, 2004, pp. 166–177.

[6] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 1–24, 2008.

[7] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, *Power sources for wireless sensor networks*. Springer, 2004, vol. 2920.

[8] J. Girao, D. Westhoff, and M. Schneider, "Cda: con- cealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 5, may 2005, pp. 3044 – 3049 Vol. 5.

[9] J. Domingo-Ferrer, "A new privacy homomorphism and applications," *Inf. Process. Lett.*, vol. 60, no. 5, pp. 277–282, 1996.

[10] J. Domingo-Ferrer and J. Herrera-Joancomart, "A privacy homomorphism allowing field operations on encrypted data," in *I Jornades de Matematica Discreta i Algorismica, Universitat Politecnica de Catalunya*, 1998, pp. 1–3.

[11] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism," in *Proceedings of the 5th International Conference on Information Security*, ser. ISC '02. Springer-Verlag, 2002, pp. 471–483.

[12] H. Lee, J. Alves-Foss, and S. Harrison, "The use of encrypted functions for mobile agent security," in *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, ser. HICSS '04. IEEE Computer Society, 2004, pp. 297–306.

[13] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," in *"book chapter of Security in Distributed, Grid, and*

[14] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in-home healthcare : Potential and challenges," *Sensors Peterborough NH*, pp. 7–10, 2005.

[15] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 51–58, february 2010.

[16] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[17] W. Hu, P. Corke, W. C. Shih, and L. Overs, "Secfleck: A public key technology platform for wireless sensor networks," in *Wireless Sensor Networks, 6th European Conference, EWSN 2009*, 2009, pp. 296– 311.

[18] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41–77, 2005.

[19] Q. Mi, J. A. Stankovic, and R. Stoleru, "Secure walk- ing gps: a secure localization and key distribution scheme for wireless sensor networks," in *Proceedings of the Third ACM Conference on Wireless Network Security, WISEC 2010*, 2010, pp. 163–168.

[20] D. Liu and P. Ning, "Multilevel μtesla: Broadcast authentication for distributed sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, pp. 800–836, November 2004.

[21] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[22] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wire- less sensor networks," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '04. New York, NY, USA: ACM, 2004, pp. 29–42.

[23] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor net- works." in *HotOS'03*, 2003, pp. 163–168.

[24] L. Lazos and R. Poovendran, "Serloc: secure range-independent localization for wireless sensor networks," in *Proceedings of the 2004 ACM Workshop on Wireless Security*, 2004, pp. 21–30.

[25] D. Westhoff, J. Girao, and M. Acharya, "Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1417–1431, October 2006. [Online]. Available: http://dx.doi.org/10.1109/TMC.2006.144

[26] L. Ertaul and V. Vaidehi, "Computing aggregation function minimum/maximum using homomorphic encryption schemes in wireless sensor networks (WSNs)," in *Proceedings of the 2007 International Conference on Wireless Networks, ICWN 2007*, 2007, pp. 186–192.

[27] M. Manulis and J. Schwenk, "Security model and framework for information aggregation in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 2, pp. 13:1–13:28, 2009.

[28] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 20:1–20:36, 2009.

[29] A. C.-F. Chan, "Concealed data aggregation for wireless sensor networks." in *Security in RFID and Sensor Networks*, Y. Zhang and P. Kitsos, Eds. CRC Press, 2009, pp. 399–416.

[30] H. Alzaid, E. Foo, and J. M. G. Nieto, "Rsda: Reputation-based secure data aggregation in wireless sensor networks," in *9th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2008*, 2008, pp. 419–424.

[31] J. Albath and S. Madria, "Secure hierarchical data aggregation in wireless sensor networks," in *Proceedings of the 2009 IEEE conference on Wireless Communications & Networking Conference*, ser. WCNC'09. IEEE Press, 2009, pp. 2420–2425.

[32] Z. Peng and Y. Jian-ping, "Secure data aggregation for sensor networks," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*. IEEE Press, 2010, pp. 1853–1856.

[33] L. Hu and D. Evans, "Secure aggregation for wireless networks," in *Workshop on Security and Assurance in Ad hoc Networks*. IEEE Computer Society, 2003, pp. 384–392.

[34] J. H. Cheon, W.-H. Kim, and H. S. Nam, "Known-plaintext cryptanalysis of the domingo- ferrer algebraic privacy homomorphism scheme," *Information Processing Letters*, vol. 97, no. 3, pp. 118–123, 2006.

[35] D. Wagner, "Cryptanalysis of an algebraic privacy homomorphism," in *Proceedings of 6th International Conference Information Security, ISC 2003*, ser. Lecture Notes in Computer Science, vol. 2851. Springer, 2003, pp. 234–239.

[36] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 188–200. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031518

[37] C. Mallanda, A. Suri, V. Kunchakarra, S. S. Iyengar, R. Kannan, A. Durresi, and S. Sastry, "Simulating wireless sensor networks with OMNeT++," Dept. of Computer Science, Louisiana State Univ., 2006.

[38] J. H. Yang, "Performance, feasibility & scalibility of hommorphic encryption schemes in securing wireless sensor networks," Master's thesis, California State University, East Bay, May 2008.

[39] C. Singh, O. Vyas, and M. Tiwari, "A survey of simulation in sensor networks," in *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, dec. 2008, pp. 867–872.

**Levent Ertaul** received B.Sc. from Anatolia University Turkey, in 1984, M.Sc. from Hacettepe University, Turkey, in 1987, and PhD degree from Sussex University, UK, in 1994. He is currently a full time Professor at California State University Eastbay, USA in the department of Math & Computer Science. He is actively involved in security projects nationally and internationally. His current research interests are Wireless Security, Ad Hoc Security, Security in WSNs and Cryptography. He has numerous publications in security issues.

**Johan Hadiwijaya Yang** graduate student in CSU East Bay, USA.

**Gokay Saldamli,** He is an assistant professor in Bogazici University, Turkey