Data Integrity Attacks and Defenses for Intel Lab Sensor Network

Renchi Yan, Teng Xu, Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{renchi.yan, xuteng, miodrag}@cs.ucla.edu

***Abstract*—Wireless sensor networks have been increasingly popular and they have been deployed in a wide range of areas including transportation system, healthcare, robotics, and smart home. Wireless sensor networks have facilitated our life using the remote sensing ability. However, these systems could be taken advantage by the malicious parties to mislead users since they are often not physically secured and are used in hostile environments. We have proposed theoretical and statistical framework for creating data integrity attacks and corresponding security defenses. The data integrity attack is defined as a type of attack that targets to mislead the system by altering some collected data of the system in such a way that to achieve the goal of attackers. We use the Intel lab sensor data and demonstrate that it is easy to attack only a few essential sensors values to change the behavior of system, such as the control of air conditioner. To be more specific, by changing the temperature and humidity data measured by a group of sensors, the attacker can control the air conditioner turning on and off without being detected by correlation check. On the top of data integrity attacks, we have further developed low overhead defense schemes.**

I. Introduction

A wireless sensor network is a collection of spatially distributed sensors that measure a set of specific characteristics of their environment in real time. Through its remote sensing ability, wireless sensor network has help people in many areas. For example, in the $novel.de$ website [1], healthcare has helped in creating numerous wireless health applications and has extended the scope of medical diagnosis. Medical experts can examine the day-to day activities of a patient or analyze the patient's physiological data collected from sensors on a regular basis. This also eliminates the need to solely rely on in-person medical check-ups while improving the quality of care. In robotics field, developing integrated wireless sensor network and robots applications could solve the problem of understanding the outside environment with noisy and imprecise sensor capabilities. In transportation systems [2], sensor is used to sense the speed of moving vehicles and to classify the vehicles according their estimated length, which could help to increase mobility, safety and passenger comfortability.

Due to the fact that wireless sensor networks are usually exposed in open and hostile environment, security has emerged to be an important issue. However, embedded sensor network enforces strict constraints on energy and power consumption. Both act as major obstacles in the application of traditional security techniques to sensors. Security is extremely critical in embedded sensor devices since even a small error can cause irreversible damage. Moreover, in order to be energy-efficient, many devices reduce the number of sensors they use. This makes them to be even more susceptible to data integrity attacks. Here the data integrity attack is defined as a type of attack that targets to mislead the system in such a way that to achieve the goal of attackers.

Many papers have been written to address communication based security issues of wireless sensor networks in embedded sensor applications [3][4]. However, we focus on data integrity attacks. A data integrity attack is the one in which the attacker modifies information in such a way that the result is incorrect, but looks correct to the casual or perhaps even the attentive viewer. We demonstrate that by implementing exceptionally simple attacks and it turns out that the attacker can hamper the result drastically by only making small changes to the data. We have evaluated our attacks and defenses on the Intel lab sensor data, which keep collecting temperature, humidity, light and voltage information every two minutes. It consists of 54 sensors located in different location in the lab. We have proved that by attacking several isolated sensors in a group, the attack can easily control the air conditioner turning on and off without being detected by normal correlation check. We also propose defenses against these data integrity security attacks that can detect our proposed data integrity attacks. We evaluate both attacks and defenses under real sensor datas.

The remainder of this paper is organized as following. We first present the related work and the preliminary knowledge in Section II and Section III. Then we show how we preprocess the data into the needed form in Section IV. Section V and Section VI provide the implementation details of our proposed attacks and defenses on sensors and evaluate the performance. Finally, we conclude the paper by summarizing our findings and stating our conclusions.

II. Related Work

In the last decade, wireless sensor devices and corresponding techniques have attracted a great deal of research interest. Various efforts have been taken to leverage the power and energy limitations of security techniques used for sensor devices, among which, many of the previous researches focus on medical devices [5][6] . In addition to security, other issues such as privacy [7][8], trust [9][10], and low power [11] have also received significant attention. As a matter of fact, many

other aspects related to security in wireless sensor applications such as systems that integrate wireless devices and cloud computing have been addressed [12][13][14].

Using the sensor data from Intel lab, researchers from University of Southern California investigated the topic that to choose a subset of sensors to minimize worst-case prediction error [15]. Researchers from Columbia University have proposed an approach to eliminate the noise from the data [16].

While all previous efforts have emphasized the vulnerabilities of used wireless security protocols and their potential fixes, we focus on actual alteration of collected sensor data in such a way that we could alter the behavior of the controller, e.g., the control of an air conditioner by compromising the integrity of the data. For medical devices, this leads to incorrect treatments. For smart home devices, this leads to wrong fire alarm. For Intel lab sensor, this leads to turning on/off the air conditioner controlled by an attacker. Specifically we show on the Intel lab sensor data how an attacker can alter only a very small number of sensor readings by a limited amount to heavily affect the behavior of the air conditioner. In addition to attacks we also have developed defense techniques that are very effective against the proposed attacks.

III. Preliminaries

A. Intel Berkeley Research Lab

The data are collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28th and April 5th, 2004.

In order to measure the whole lab environment precisely, 54 sensors are roughly distributed everywhere in the Research lab. Those sensors collected timestamped topology information, along with humidity, temperature, light and voltage values in nearly every two minutes. We have in total 2.3 million readings collected from these sensors.

B. Data Set

The data is collected from 54 sensors deployed in the Intel Berkeley research lab between February 28th and April 5th, 2004. In order to measure the whole lab environment precisely, 54 sensors are roughly distributed everywhere in the lab. Those sensors collected timestamped topology information, along with humidity, temperature, light and voltage values with the frequency of every two minutes. We totally have 2.3 million readings collected from these sensors.

Our dataset includes temperature, humidity, light and voltage over all of the 54 sensors. Temperature is in degrees Celsius. Humidity is temperature corrected relative humidity, ranging from 0-100%. Light is in Lux (a value of 1 Lux corresponds to moonlight, 400 Lux to a bright office, and 100,000 Lux to full sunlight). Voltage is expressed in volts, ranging from 2 to 3.

IV. Preprocessing

Before we do the attack and defense, we need to preprocess the data. There are two major challenges in the preprocessing. Firstly, for each single sensor, there are some missing epochs in the data set. This means the data may not be consecutive. For example, sensor 1 measured the data at April 5th 1:00:02:21, and the next measurement occurred on April 5th 1:00:08:43. Due to the fact that sensor observations provided in different timestamp formats, we preprocessed and unified timestamp formats in our experiments.

Our approach is to split the days into time slots. Each time slot has two minutes. If sensor i has more than 1 measurement during that time slot, we take the average of those measurements. If sensor i has no measurement on time slot t_1, we go backward and find the first time slot t_2 ($t_2 < t_1$) that has measurement. Then we go forward and find the first time slot t_3 ($t_3 > t_1$) that has measurement and calculate the weighted average for t_1. The equation is shown in equation 1. T_{i,t_1} represents the temperature for sensor i at time slot t_1. The whole algorithm is shown in Algorithm 1.

$$T_{i,t_1} = [T_{i,t_2} * (t_3 - t_1) + T_{i,t_3} * (t_1 - t_2)]/(t_3 - t_2) \quad (1)$$

Algorithm 1 Preprocessing

Input: D - Data set
Input: N - number of sensors

1: **for** 1 <= i <= N **do**
2: **for all** `data in` D_i **do**
3: `Save into time slot` $T_{i,t}$
4: **end for**
5: **for all** `t1 in time slot` **do**
6: **if** `their is more than one measurement in` $T_{i,t}$ **then**
7: `Take the average`
8: **end if**
9: **end for**
10: **for all** `t1 in time slot` **do**
11: **if** $T_{i,t1}$ `== empty` **then**
12: `go backward, find the first time slot t2 that` $T_{i,t2}$ `!= empty`
13: `go forward, find the first time slot t3 that` $T_{i,t3}$ `!= empty`
14: $T_{i,t1}$ `= [`$T_{i,t2}$ `* (t3-t1) +` $T_{i,t3}$ `* (t1-t2)] / (t3-t2)`
15: **end if**
16: **end for**
17: **end for**
18: `Output:` T

V. Security Attacks

A. Goals and Challenges

There are various types of attacks can be applied on the wireless sensor networks. Our goal is to mislead the central air conditioner as much as possible. The central air conditioner will open if the temperature of any sensor is above a certain threshold or below a certain threshold. We target to control the

air conditioner by changing the reading of a certain number of sensors by a certain amount at a specific time slot.

However, the main challenge is to maintain a balance between the outcome of the attack and the risk of the attack being detected. For example, the attacker can change the temperature data of many sensors by a large percent. In this case, it is hard for the attacker to get access to such a big number of sensors. On the other hand, as illustrated in following paragraph, the attack will be easily detected if many sensors show extremely abnormal values. As a result, the last criteria in designing data integrity attacks is to assume limited access for the attacker and try not to be obviously suspicious to medical experts.

In the dataset, different pairs of sensors have different correlations with each other. Figure 1 shows the humidity comparison between sensor 50 and sensor 16 and Figure 2 shows the comparison between sensor 50 and sensor 51. Since sensor 50 and sensor 51 are close in location, their humidity shows a much stronger correlation. In Figure 3 and Figure 4, we respectively show the overall temperature and humidify correlation between all sensor pairs using colormap.

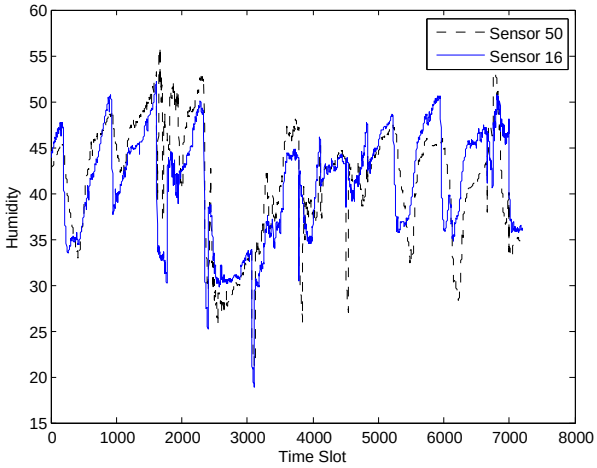

Fig. 1: The humidity of sensor 50 and sensor 16.

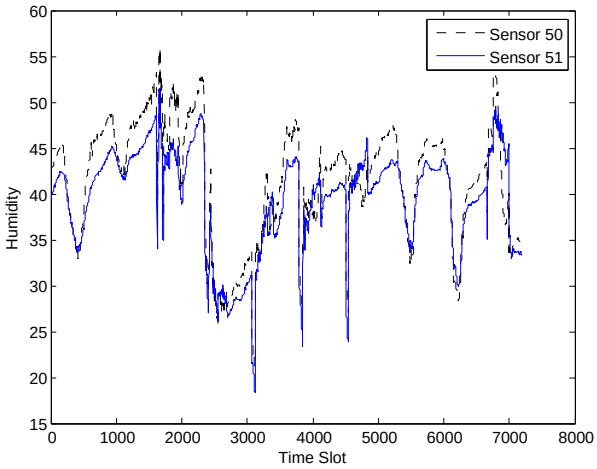

Fig. 2: The humidity of sensor 50 and sensor 51.

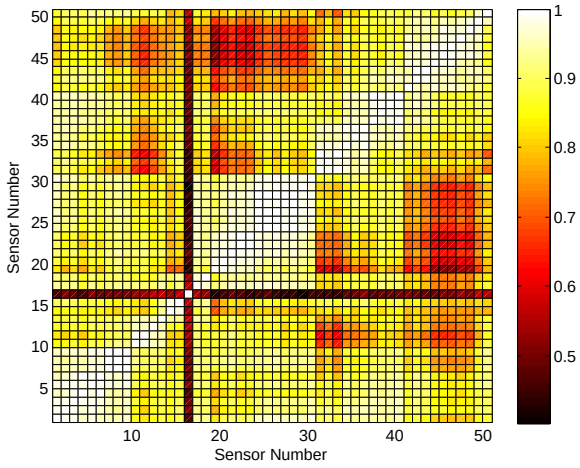

Fig. 3: The correlation of temperature for all pairs of sensors.

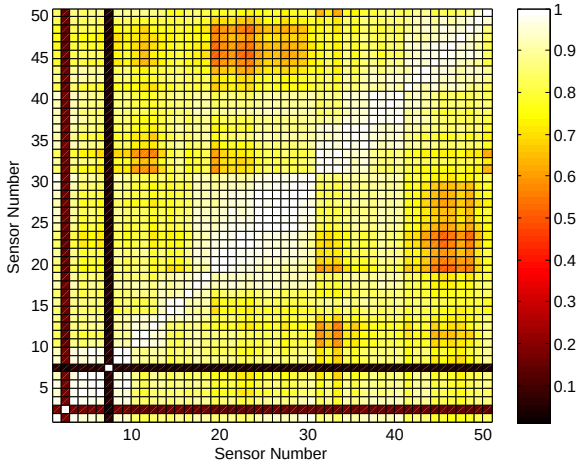

Fig. 4: The correlation of humidity for all pairs of sensors.

For a specific sensor i, the sensors that are top correlated with i may as well vary from time to time. For example, for a specific sensor 30, its temperature data during time period 8:00AM to 9:00AM is mostly correlated with sensor 26, 31 and 32 with correlation 0.9947, 0.9944 and 0.9934. However during 5:00PM to 6:00PM, the top three correlated sensors change to sensor 31, 29 and 32 with corresponding correlation 0.9928, 0.9925 and 0.9898. Thus, we need to find specific time periods to as well as specific sensors to attack.

In principle, if a sensor is not well correlated with all the other sensors in a period of time, it is the best target to attack. However, since the sensors in the lab are located evenly across the space, it is hard to find only a single sensor to attack. Instead, our goal is to find a small group of "isolated" sensors as the target for attack. An isolated group of sensors G is defined as the following. For each sensor s_i, it has a group of sensors g_i, where the sensors in g_i are the top n correlated sensors to k_i. For each sensor in G, it must be included in every g_i of all the other sensors in G.

B. Attack Modeling

We define the $C_{i,k,t}$ as during time period t, the average of top k correlation between sensor i and all the other sensors. For example, if k is 3, for sensor 30, temperature data during timer period 8:00AM to 9:00AM is mostly correlated with sensor 26, 31 and 32 with correlation 0.9947, 0.9944 and 0.9934. Thus, $C_{30,3,8-9}$ is 0.9941. Our goal is to find the isolated group during a time period that has the minimum average $C_{i,k,t}$ for all sensor i in the isolated group.

$$\begin{aligned} &\text{Minimize } Avg(C_{i,k,t}) \\ &\text{Subject to} \\ &\qquad n \leq N \\ &\qquad k \leq K \\ &\qquad t \leq T \\ &\text{where} \end{aligned} \tag{2}$$

- n is the number of sensors in the isolated group.
- $C_{i,k,t}$ for all sensor i in isolated group.
- T is the Max time period to attack.
- N is the number of sensors attacked.
- N, K, T are constants.

The core idea is to use breadth first search to solve this problem. At a specific time period t, first we calculate the correlation for all pairs of sensors. Then for each sensor i, we find top k most correlated sensors. We define sensor i is interested in sensor j if sensor j is among the top k most correlated sensors of sensor i. We build a graph based on the correlations. Each sensor is a node in the graph. Initially there is no edge in the graph. If sensor i is interested in j and sensor j is interested in i, we connect an edge between sensor i and j. Then we do the breadth first search. We enumerate the start point for each sensor i, and all the sensors that are reachable from sensor i are included into this isolated group. Finally we pick the isolated group based on $Avg(C_{i,k,t})$. The overall algorithm is shown in Algorithm 2. Once we get the optimal isolated group, then during a certain time period, we will freeze those sensors in the group and let those sensors keep recording the last correct data point instead of new measurements. For example, if sensor 16 and 17 are chosen to be attacked during 8:00AM to 10:00AM, they will keep recording the value that they measured at 7:58AM. So if the attack is in the morning, the air conditioner will keep heating the room because of our attack, even if the measurements of temperature from all other sensors is high enough to close the air conditioner. If the attack is in the afternoon, the air conditioner will keep cooling the room because of our attack, even if the measure of temperature from all other sensors is low enough to close the air conditioner.

K	Isolated group	Time period	$Avg(C_{i,k,t})$
3	Sensor 16,17	8:00AM-11:00AM	0.7061
3	Sensor 16,17	8:00AM-10:00AM	0.7219
3	Sensor 16,17	8:00AM-12:00PM	0.7280
3	Sensor 16,17,35,36	8:00AM-9:00AM	0.8358
3	Sensor 50,51	9:00AM-11:00AM	0.8407

TABLE I: Optimal isolated group to attack when $k = 3$.

Algorithm 2 BFS finding optimal isolated group

Input: N - Number of sensor
Input: k - Constant

```
for Each time period t do
  calculate correlation for all pairs of sensor.
  for 0 <= j < N do
    Sort the correlation of all sensors to sensor j
  end for
  Construct the graph.
  for 0 <= j < N do
    Find all sensors that is reachable from sensor j, form the isolated group.
    Calculate the Avg(C_{i,k,t})
  end for
end for
Return the best isolated group.
```

K	Isolated group	Time period	$Avg(C_{i,k,t})$
4	Sensor 16,17	8:00AM-11:00AM	0.6698
4	Sensor 16,17	8:00AM-10:00AM	0.6907
4	Sensor 16,17	8:00AM-12:00PM	0.6992
4	Sensor 16,17,35,36	8:00AM-10:00AM	0.7818
4	Sensor 16,17,34,35,36	8:00AM-9:00AM	0.7917

TABLE II: Optimal isolated group to attack when $k = 4$.

K	Isolated group	Time period	$Avg(C_{i,k,t})$
5	Sensor 16,17	8:00AM-11:00AM	0.6435
5	Sensor 16,17	8:00AM-10:00AM	0.6543
5	Sensor 16,17	8:00AM-12:00PM	0.6814
5	Sensor 16,17,34,35,36	8:00AM-9:00AM	0.7428
5	Sensor 20,14	8:00AM-9:00AM	0.7651

TABLE III: Optimal isolated group to attack when $k = 5$.

K	Isolated group	Time period	$Avg(C_{i,k,t})$
6	Sensor 16,17	8:00AM-11:00AM	0.6184
6	Sensor 16,17	8:00AM-10:00AM	0.6263
6	Sensor 16,17	8:00AM-12:00PM	0.6616
6	Sensor 16,17,34,35,36	8:00AM-9:00AM	0.6967
6	Sensor 16,17,35,36	8:00AM-11:00AM	0.7319

TABLE IV: Optimal isolated group to attack when $k = 6$.

K	Isolated group	Time period	$Avg(C_{i,k,t})$
7	Sensor 16,17	8:00AM-11:00AM	0.5997
7	Sensor 16,17	8:00AM-10:00AM	0.6463
7	Sensor 16,17,34,35,36	8:00AM-9:00AM	0.6553
7	Sensor 16,17	8:00AM-13:00PM	0.6827
7	Sensor 16,17,36	8:00AM-12:00PM	0.7134

TABLE V: Optimal isolated group to attack when $k = 7$.

C. Evaluation

We run the algorithm on different k. For each situation we list the top isolated groups to attack and their corresponding $Avg(C_{i,k,t})$. The result is showed in table I, II, III, IV, V.

First of all, there are some cases where for different k, the same isolated group is chosen to be attacked. For example,

the sensor 16 and 17 during time period 8:00AM-11:00AM, the $Avg(C_{i,k,t})$ keeps changing with a different k value. This is because the difference in k leads to the difference in $Avg(C_{i,k,t})$. As illustrated in the last subsection, the definition of $C_{i,k,t}$ is that during time period t, the average of top k correlation between sensor i and all the other sensors. Therefore, if we choose the same isolated group during the same time period with different k, the $Avg(C_{i,k,t})$ will be different.

Another observation is that, a higher k will lead to a larger isolated group. Based on our algorithm, this is reasonable. When we construct the graph, the higher k will lead to a higher connectivity graph which result in larger isolated groups. For example, in Table I, the isolated group during 8:00AM-9:00AM has 4 sensors (16,17,35,36). But it is expanded to 5 sensors(16,17,34,35,36) in Table II. In Table IV the isolated group during 8:00AM-12:00PM has 2 sensors(16,17). It increases to 3 sensors(16,17,36) in Table V.

VI. DEFENSE

We propose a corresponding technology to possibly detect and analyze data integrity security attacks in this section.

A. Goals and Challenges

Two essential goals are addressed for the defense technology. The first one is to detect, diagnose, and also to remove the impact of attacks as much as possible. The second is to design the defense to be low-cost, low-energy and real-time which is due to the requirements of the wireless sensor networks. As a result, traditional cryptographic methods are not suitable in this scenario; customized methods need to be proposed.

The main challenge in defense technology is to find out whether the data abnormality is because of the weather itself or caused by the data integrity attacks. Assume that we check the data on the sensor and find the humidity is extremely high, it is possible that there is a rainstorm outside the lab which happens once or twice every year. It is also possible that the data is attacked. The above challenges are taken into the consideration of our suggested defense.

B. Defense Procedures

As we described in the attack section, we freeze the sensors in the isolated group during the attack period. This means the measurement for each sensor will keep the same during the attack period. This will lead to the result of those sensors being attacked have 100 percent correlation with each other. If we run algorithm 2 based on the attacked data, those attacked sensors will be selected as isolated group. So, in order to detect whether the data has been attacked, we run the algorithm 2 first to find a list of suspects.

After we get the suspects list, for each isolated group, we split the sensors in the original group into two subgroups. And then we compare the weighted average correlation of the optimal splitting plan with the weighted average correlation of the split plan that splits each hour as a single time period. We define δ as the difference between the above two. We found that the δ of an isolated group will change in a significant amount if we attack that group. Thus, we can use the variation of δ to detect attack.

C. Evaluation

We check the variation of δ based on original data and different attacked data. We show some cases in table VI. In general, if the attack persists longer, the change of δ will be more significant.

Isolated group	Time period	Abs(variation of δ) %
Sensor 16,17	8:00AM-11:00AM	14.9
Sensor 16,17	8:00AM-12:00PM	19.5
Sensor 16,17,35,36	8:00AM-9:00AM	20.4
Sensor 16,17,34,35,36	8:00AM-9:00AM	18.8
Sensor 50,51	9:00AM-11:00AM	17.9

TABLE VI: The variation percentage of δ (absolute value) after attack

VII. CONCLUSION

In this paper, we have proposed the data integrity attacks on the Intel Berkeley lab sensor data. Our results have indicated that by only changing a small subset of sensors, there is high likelihood that the controls on the air conditioner will be altered. We have also proposed defense approaches which targets to detect the data integrity attacks. The evaluation on the defense also shows accurate detection by using the variation percentage of parameter δ.

VIII. ACKNOWLEDGEMENT

This work was supported in part by the NSF under Award CNS-0958369, Award CNS-1059435, and Award CCF-0926127, and in part by the Air Force Award FA8750-12-2-0014.

REFERENCES

[1] Novel.de, Pedar, *http://www.novel.de/*, 2007.
[2] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless communications and mobile computing*, vol. 9, no. 3, pp. 287-302, 2009.
[3] H. S. Ng, M. L. Sim, and C. M. Tan, "Security issues of wireless sensor networks in healthcare applications," *BT Technology Journal*, vol. 24, no. 2, pp. 138-144, 2006.
[4] M. A, Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of medical systems* vol. 36, no. 1, pp. 93-101, 2012.
[5] R. Yan, V. C. Shah, T. Xu and M. Potkonjak, "Security Defenses for Vulnerable Medical Sensor Network," *International Conference on Healthcare Informatics*, pp. 300-309, 2014.
[6] T. Xu, J. B. Wendt, and M. Potkonjak, "Matched Digital PUFs for Low Power Security in Implantable Medical Devices," *International Conference on Healthcare Informatics*, pp. 33-38, 2014.
[7] X. Lin , R. Lu , X. Shen , Y. Nemoto and N. Kato, "SAGE: a strong privacy-preserving scheme against global eavesdropping for ehealth systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 365-378, 2009.
[8] T. Xu, H. Gu, and M. Potkonjak, "Data Protection Using Recursive Inverse Function," *International Conference on Field Programmable Logic and Applications*, 2015.
[9] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT Systems: Design Challenges and Opportunities," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 417-423, 2014.

[10] T. Xu, D. Li, and M. Potkonjak, "Adaptive Characterization and Emulation of Delay-based Physical Unclonable Functions Using Statistical Models," *Proceedings of the 52nd Annual Design Automation Conference*, pp. 76-81, 2015.

[11] T. Xu, and M. Potkonjak, "Energy Saving using Scenario based Sensor Selection on Medical Shoes," *International Conference on Healthcare Informatics*, 2014.

[12] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing," *International Journal of Security and Networks*, vol. 6, no. 2, pp. 67-76, 2011.

[13] L. Fan, W. Buchanan, C. Thummler, O. Lo, A. Khedim, O. Uthmani, A. Lawson, and D. Bell, "Dacar platform for ehealth services cloud," *IEEE International Conference on Cloud Computing (CLOUD)*, pp. 219-226, 2011.

[14] T. Xu, and M. Potkonjak, "The Digital Bidirectional Function as a Hardware Security Primitive: Architecture and Applications," *ACM/IEEE International Symposium on Low Power Electronics and Design*, 2015.

[15] A. Das, and D. Kempe, "Sensor selection for minimizing worst-case prediction error," *Information Processing in Sensor Networks*, pp. 97-108, 2008.

[16] P. Ji, and M. Szczodrak, "A multivariate model for data cleansing in sensor networks," *The Second Annual Conference of the International Technology Alliance*, 2008.