Spring 3-20-2020

Demonstrating the Threat of Hardware Trojans in Wireless Sensor Networks

Maryam Jalalitabar

Marco Valero

Anu G. Bourgeois

Demonstrating the Threat of Hardware Trojans in Wireless Sensor Networks

Maryam Jalalitabar Marco Valero Anu G. Bourgeois
Department of Computer Science
Georgia State University
Atlanta, Georgia
mjalalitabar1@student.gsu.edu mvalero@cs.gsu.edu abourgeois@cs.gsu.edu

***Abstract*—As the demand for cheaper electronic devices has increased, the location of manufacturing foundries has changed, sometimes to untrusted places in foreign countries. Some of these locations have limited oversight of the manufacturing of complicated and sensitive electronic components including integrated circuits (ICs). The integrated circuits are key component in all current electronic devices and can be modified to be malicious or to monitor the functions of their applications. These malicious modifications on the ICs are called hardware trojans (HWTs). HWTs can be designed to quietly monitor, to actively send out sensitive information, or to destroy their host device completely. The idea of hardware trojans in Wireless Sensor Networks (WSNs) has not been investigated before; thus, our goal is to demonstrate the potential threat that hardware trojans pose for sensor networks. This is important to study, given that in WSNs hundreds of sensors are deployed and in most cases left unattended, which gives the opportunity to an attacker to trigger a HWT on the sensors. For our investigation, we used TelosB sensors that have been used for some WSN applications. An attacker in a network can, for example, take advantage of the SPI bus that is used by the radio to eavesdrop messages and even disrupt communications completely. Currently, security breaches through software is given great importance in the WSN academic and research community. Our research shows that the same level of importance must be given to attacks through hardware to ensure a trusted and secure network.**

KEY WORDS: WSN, hardware trojan, TelosB.

I. Introduction

Wireless sensor networks (WSNs) are being used widely for different purposes such as military, health, and environmental monitoring. In military applications, sensors are used for target tracking, surveillance, intrusion detection, or identification [1], [2]. Other applications are flood alerting system [3], biomedical health monitoring and volcano monitoring [4], [5], [6]. In these applications, a large number of sensors construct a WSN. Each of these devices consists of a sensing unit, processing unit, transceiver unit and a power unit. The sensing unit includes different sensors such as light, temperature and humidity. The processing section is for basic computational tasks. The transceiver is the singular way for the sensors to communicate with each other. As for the power unit, sensors are equipped usually with small batteries.

Among the various features of WSNs, the rapid deployment, self-organization and fault tolerance characteristics along with the small size and low cost make them a highly effective sensing facility to be utilized in different applications. In WSNs, the number of the sensors that are deployed to monitor an incident can reach thousands [7]. With this scale, loss of a number of these sensors do not influence the operation of the network and provides a level of fault tolerances to the network.

In some of these applications, the data collected by the sensors is sensitive and private. For example, in target tracking for the battlefields, data should be encrypted and submitted securely to the end users using an encryption method. Here, as the sensors are working with delicate information, any threat to the functionality of the sensors can be critical. Some other applications are dependent on timely delivery of the data. For instance, in a fire detection sensor network for a forest, if there is a fire, information that is collected by the sensors should be received at the monitoring station in time for decision making process. In such an application, if the data transmission is interrupted or delayed, it may cause devastating consequences.

Some WSN applications require the sensors to be in proximity of the event. In these cases, sensors are designed to accomplish their tasks in unattended working areas. Examples of such places include battlefields, large warehouses and forests. Data that is gathered by the sensors in these applications can be confidential or time sensitive. Thus, to guarantee the safety of the sensors, security considerations must be addressed with high priority.

There are different types of security attacks in the sensor networks, including node replication attacks [8], DoS attacks [9], attacks on the privacy of the network such as eavesdropping and passive monitoring [10], and many others. However, as we will show in this paper, there is another security threat to the safety of the sensor networks that has not been addressed before.

In particular, we will describe the potential threat of Hardware Trojans (HWTs) in sensor networks. A HWT is a deliberate modification of the hardware during the fabrication process, with the intent to impose a major threat to the functionality of the networks. This has been widely studied for traditional networks, but not yet considered in the context of wireless sensor networks, where devices can be left unattended. If a sensor device has been tampered by a HWT, they can alter performance by corrupting data and/or interrupting communication. Our paper focuses on establishing

the feasibility of this type of threat in WSNs. It also serves to encourage the community to address security measures starting from the fabrication process of the sensors.

In order to demonstrate the threat of HWTs in a sensor network, we have conducted multiple experiments using off the shelf products. The experiments illustrate the potential impact of HWTs on a sensor network comprising TelosB sensors. We exhibit how an attacker can activate a HWT by tampering with the TelosB hardware board. As we do not have access to the fabrication process, our assumption is that a HWT has been inserted in the sensor board.

This work is important due to the dearth of knowledge on the subject. Currently, security breaches through software is given great importance in the WSN academic and research community. Our research shows that the same level of importance must be given to hardware based attacks to ensure a trusted and secure environment.

The rest of this paper is organized as follows: Background is explained in Section 2. In Section 3, we go over the literature review. Section 4 explains the system model and experiments. Finally conclusion and future work are expressed in Section 5.

II. BACKGROUND

There are a number of obstacles that cause security concerns to be different in WSN from the conventional networks. Consequently, existing security techniques are not completely adequate or applicable for the sensor networks. Some of these obstacles are:

A. Limitations of WSNs

1) ***Restricted Resources****:* In order to apply a security mechanism, a specified extent of hardware and software assets must be available which include memory space and power resources. But sensors that operate in WSNs are strictly limited in terms of memory size and power consumption. For instance, a typical sensor type like TelosB has 10KB of RAM memory and it is powered by a small pack of batteries. Thus, the security codes that are designed to be installed on a sensor must be small in size. Energy constraint is another important concern in WSNs. As the sensors are deployed in a network, they are expected to operate for long duration of time and to apply a security method for a sensor, the energy burden must be taken into consideration.

2) ***Wireless Communication****:* The security of the network relies heavily on a defined protocol, which in turn depends on communication [11]. Considering the communication medium for the majority of sensor networks is wireless, WSNs are prone to security attacks. Packets that are submitted via wireless channels are at risk of being dropped or damaged. Also, when using wireless medium, there is a high risk of collisions due to fact that all nodes broadcast on a shared medium and in a highly condensed WSNs, failure of packet submission is prevalent [12].

3) ***Unattended Operation****:* As mentioned earlier, based on the type of the application that WSNs are used for, sensors might be expected to operate in an unattended environment for a long time. This means that the sensor nodes are exposed and vulnerable to physical attacks. If the habitat is easily accessible, then an adversary can simply tamper with the devices. In this situation, the likelihood that a sensor deteriorates is much higher than devices deployed in secured network, like a typical personal computer (PC) in a network.

As the WSN is a unique type of wireless network with its own special limitations, the security requirements for such a network are also different. In order to make data communication safe, data must be encrypted and the secret key should be accessible by the receiver only. If that is the case, data confidentiality is attainable. However, even with the implementation of data encryption, a sensor network is still not safe completely. It is possible for an adversary to disturb or destroy the sensors' communication.

B. Hardware Trojans

The issue of trust is an emerging problem in integrated circuit (IC) security [13]. High demand for reducing the costs due to economic concerns, has pushed the ICs manufacturing to offshore foundries. As a result, the control over the process of IC fabrication has been reduced significantly and they are more susceptible to malicious attacks by adversaries. Schematics and plans are sent and we simply trust those foundries to manufacture and deliver the desired product. This leaves the IC fabrication process prone to hardware trojans. HWTs are the malicious altering of hardware specifications or implementation in such a way that its functionality can be modified under a set of conditions defined by the attacker [14].

The set of conditions, are the triggering events that provoke the HWT on the IC. Triggering the HWT can be performed in different ways such as remotely or by direct physical access. HWT can be enabled remotely, as the device is not physically accessible, by a specific input event. The HWT could be triggered by a particular sound or light pattern. In this case, an adversary does not need to physically access the device, but can trigger it remotely.

For example, a HWT can enforce the chip to leak out information covertly in different ways such as optical, thermal or radio. In optical type of HWT, a LED on the circuit is electrically adjusted in a way that it blinks at a rate which is indistinguishable by human eye. This optical signal can be accessed using an optical to audio amplifier. Thermal trojans cause an external resistor on the chip to emit heat. Different parts of the chip, such as the microcontroller are saturated, they will create thermal heat. An adversary can then receive this thermal signal using an infrared camera [15].

If the attacker can get physical access to the device, he can provoke the HWT directly. For instance, an attacker can use a particular input string which is inserted via a less secured section of the IC such as unused input pin.

Most of the testing methods for HWT detection assume the existence of a golden IC that is obtained through a complete

testing process of a random selected IC. The IC that has been completely cleared from validation tests will be called the golden IC [16]. After obtaining the golden IC, it will be the trusted resource for checking the other ICs. The mechanism of attaining the golden IC is hard, complicated and it needs professional and expensive testing tools. In case of WSNs, since some of the sensor networks are less secured, it is possible for an adversary to activate a HWT on the sensors. That is, an attacker can gain control of the sensors after deployment in the network and activate the HWT. Thus the threat of the HWTs in wireless sensor networks must be addressed. Considering we do not have the proper accessories to test our TelosB sensors, we can not verify if the sensor is carrying a HWT or not. Thus, for the purpose of our experiments, we assume that our sensors are contaminated with the HWTs.

Let us consider the case that sensors are manufactured in an untrusted factory in a foreign country and due to lack of supervision in the fabrication process HWT has been inserted to the sensors. These HWTs are extremely hard to detect as they are designed to escape validation tests and would still be able to destroy the chip or leak sensitive information [17]. Later these infected sensors will be cleared from validation tests and they will be ready to be spread out in the network to accumulate data from the environment. If the desired sensor network is not highly secured and sensors will be left unattended in the sensing area, it is possible for an adversary to get physical access to the sensors and activate the HWT on the sensors which can cause major effect on the functionality of the sensor. For example, sensors can be deployed in the battlefields for target tracking. Thus, the information that is gathered by the sensors is sensitive and must be delivered on time to the base station. An adversary can activate the HWT on the sensors to stop them from transmitting the data or to destroy them completely.

III. Literature review

In this section, we review some previous works on the security attacks in WSNs. Also, due to the importance of HWT, we review the current works in this area. Sensor networks are vulnerable to different types of security attacks because of their distributed nature and their deployment in hard to access environments. These threats can make a huge effect on the functionality of WSNs. According to [9], some of the major security attacks in WSNs are:

1) ***Attacks on secrecy and authentication****:* These attacks target the standard cryptographic techniques that protect the secrecy and authenticity of sensors' communication. Attacks under this category include node replication attack and attacks on privacy such as *eavesdropping and passive monitoring*.

Parno et al. [8] investigated node replication attack in which the attacker tries to insert a node to the sensor network by copying or replication of the node identifier. Authors designed algorithms for distributed node-replica detection.

Eavesdropping and passive monitoring is one most common attacks on the privacy of the data in the sensor network. If the information is not encrypted well, an adversary in the network can eavesdrop the message. Dai et al. [10] proposed a model to inspect the probability of the eavesdropping in single hop and multihop sensor networks. They found that using directional antennas in these networks can reduce the eavesdropping chance significantly.

2) ***Attacks on network availability****:* These type of attacks are often called denial-of-service (DoS) attacks. DoS attacks can affect different layer of the network. One type of DoS attacks in the physical layer is the jamming attack. It interferes with the radio frequency of the nodes in the network [9], [18]. For the network layer, some of the attacks include sinkhole [19] and sybil [20].

There are other works on the security attacks in WSNs [21]–[23], but as far as our knowledge, the idea of HWT threat in wireless sensor networks has not been investigated before.

As the concerns about the HWTs has increased due to extensive outsourcing of the IC manufacturing process to untrusted foundries, many researchers have worked on the topic of HWTs.

Wang et al. [24] provided one of the first exhaustive studies on the topic of hardware trojans. They investigated different possibilities for malicious tampering of ICs. Also, they proposed a framework to classify different types of HWTs. Moreover, they examined some HWT detection methods for most eminent types of hardware trojans.

A specific class of HWTs which is called Trojan side channels (TSC) was introduced by Lin et al. [25]. They devised a new type of hardware trojan, which is responsible for developing artificial power side-channels. TSCs are appropriate to secretly leak private information. They can submit the data using side channel signals. An attacker who has implemented the TSC can receive and decode the information.

There are other works in the literature focused on HWT implementation and detection [26], [27], [28]. Reviewing previous works on the security attacks in sensor networks and the research on the HWT, makes it clear that HWTs threats have been addressed and investigated for different types of ICs but none of the existing works address the threat of hardware trojans for the sensor devices in WSNs. Thus, in this paper for the first time we investigate the feasibility of the HWTs in the sensor networks.

IV. System Model and Experiments

In this section, we describe the system model and the particular experiments that we have conducted.

A. System Model

We consider a system including TelosB sensors as the network and an attacker device to play the role of an adversary who tries to infiltrate the network by activating a HWT on a sensor. TelosB or Tmote Sky sensor is an ultra low power wireless module for use in monitoring applications [29]. Chipcon CC2420 radio is embedded for wireless communications on the TelosB. The radio communication is controlled by the TI MSP430 microcontroller through the SPI bus.

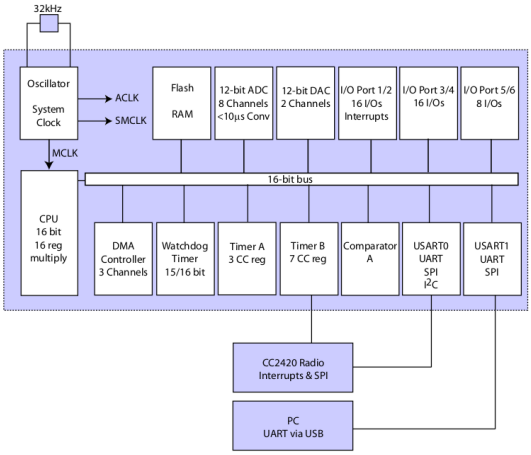

Fig. 1. MSP430 Block Diagram.

Serial Peripheral Interface or SPI is a simple 4-wire serial bus interface that is used mainly by the microcontroller to establish communication with peripheral devices. In a SPI communication, there is always a master and one or more slave devices. The master is the one that starts the communication and controls the communication. Once the communication is established, data can be transmitted from both sides, as the connection is full-duplex. The standard SPI protocol has 4 signal wires: MOSI, MISO, SCK and SS. In Figure 1, the block diagram of the MSP430 on TelosB and its connection to peripherals is shown. Another device that we used in our experiments is the Arduino, which is a microcontroller board based on the ATmega328 [30].

B. System Implementation

To prove our concept of view, we deployed a sensor network and we tried to demonstrate what will happen if an attacker can tamper with the sensors which we assumed carry a HWT.

We implemented a system consisting the TelosB sensors and an attacker device. This attacker is used to prompt the HWT on the sensors in the network. It should be mentioned that as we do not have tools to validate if the sensors have been contaminated with the HWT, we create a similar scenario. That is, we attempt to destroy the functionality of the sensors and make them behave in abnormal way which is equivalent to activating the presumed HWT. As we explained earlier, when a HWT is activated on a device, it forces the system to show unusual behavior. To implement our idea, we investigated different methods to find a way to breach the sensor hardware.

The radio is the only communication method for the wireless sensors. Our goal was to disable the radio communication of the TelosB. We focused on interrupting the radio communication of the sensors by exploiting the SPI bus on the sensor. As depicted in the Figure 1, the USART0 on the MSP430 provides the SPI bus functionality for the TelosB sensor. When the sensor wants to transmit data via radio, the SPI bus will be used. Also, if the sensor wants to establish communication with a peripheral device, this will happen through the SPI bus.

Thus, SPI is used for two purposes on the TelosB, which should not interfere with each other. It means that when the TelosB is communicating with other sensors, it should not be able to make SPI communication with another device. Otherwise, the radio communication will be interrupted and the sensor that is supposed to transmit information can not send or receive data. Also, if the data is transferred over the SPI bus, it should be protected via encryption so an adversary can not sniff the data packet during radio communication of the sensors in the network.

We exploited the SPI bus to interrupt the normal functionality of the sensor and make it behave viciously which is similar to the case that a real HWT is activated on the sensor by an attacker who has physical access to the sensor.

A group of TelosB sensors constitute the WSN. To get the sensors to be actively sending and receiving messages over the radio, they run an application called **RadioCountToLeds**. In this application, each mote will act as a 4Hz counter that broadcasts its counter value each time that it gets updated. The other node that receives a value as a packet, displays the number of the counter on its LEDs. In this scenario, radio chips of the sensors are actively involved as the wireless communication between the TelosBs is running via radio.

Our goal is to break off the radio communication between the sensors by using an attacker device. This is done via exploiting the SPI bus of the TelosB. If the attacker can disrupt the communication of the sensors and make them behave in abnormal way, we can prove our concept that it is possible to activate the HWT on the sensors in the field. To stop the sensor from transmitting data, we have to find a way to the disable the SPI communication between the CC2420 (radio) and MSP430 (microcontroller). This is because on the TelosB radio communication is controlled by MSP430. We use the attacker to demolish the SPI communication between the CC2420 and MSP430. By use of the attacker we can generate a source, which is a signal, to manipulate the SPI bus.

C. Experiments

We have conducted different types of experiments. The common base in all of these experiments is the use of the Arduino as the attacker device. The difference between the models is the way that we generate the fake source to interrupt radio communication of the sensors.

For the first experiment, our goal is to break the sensors' communication by using a specific sequence of binary numbers. In the second attempt, we exploit a PIR sensor to disturb the TelosBs' transmission. As another method for the attack, we use *Morse code* to be able to submit meaningful patterns. Morse code is a communication method in which the text information will be transmitted as tones, lights, or clicks in an on-off pattern. The message can be understood by a professional listener without any special equipment.

Binary pattern, PIR sensor, or Morse code, if they will be applied directly to the different pins on the TelosB such as MSP430 microcontroller (MCU) pins, they will not be sufficient without exploiting the SPI bus. Thus, we establish

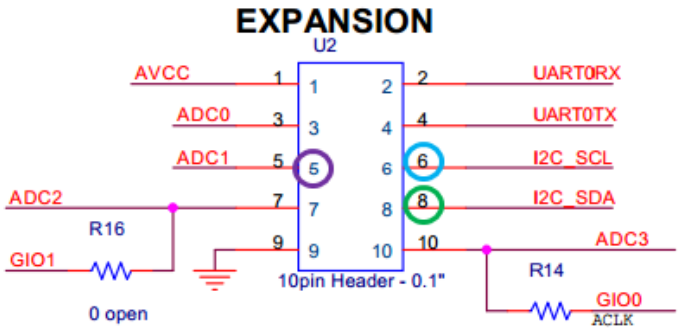

Fig. 2. TelosB U2 expansion header.

SPI communication between the Arduino as a peripheral device and TelosB.

Arduino which is used as the basic part of the attacker system, is designed with 14 pins as digital input/output. Each of these pins can be used either as input or output. Some of the pins also provide special functions. For SPI communication with other devices via the SPI library the following pins are used:

- 10 (SS- Slave Select) - The master uses to enable/disable Slave,
- 11 (MOSI- Master Out Slave In) - Sending data to the peripherals,
- 12 (MISO- Master In Slave Out) - Sending data to the master,
- 13 (SCK - Serial Clock) - Synchronizes data transmission and is generated by the master.

On the TelosB, three wires of the SPI 4-Wire bus are accessible via U2 expansion header on the TelosB board which is depicted in Figure 2. These are SS, MOSI and SCL. The last SPI pin will be directly attached to MISO pin on the MCU. It should be noted that since our goal is to break the radio communication, we do not need to transfer data on MISO from the TelosB to the Arduino. That is, there will be no data sent from slave which is the telosB to the master which is Arduino.

In all of the following experiments, it is the responsibility of the Arduino device to establish the SPI communication with TelosB and transfer the signal, that is generated by different sources, to the sensor which in turn triggers the TelosB to stop its normal operation.

*1) **Attack using a Binary pattern**:* In the first experiment, we generate a binary pattern to be transmitted on MOSI line from the master that is Arduino to the slave TelosB. In parallel, sensors in the network have their normal operation which can be seen as the LEDs of the sensors are broadcasting the current amount of their counter.

The SPI communication is formed between one of the sensors and Arduino. Data on the MOSI line will be used to trigger and stop the radio communication between the sensors. When the current bit in the binary sequence is equal to one, we enable the SPI connection. This input will be carried over the SPI data line MOSI. As soon as the data is received by the MOSI line on the TelosB, radio communication of the sensors will be stopped and LEDs will stop blinking. When the current data in binary pattern equals to zero, it will cause the SPI communication to be terminated and sensors would resume radio transmission.

Also, to better visualize the current value of the binary pattern, we use a red LED that is connected to the Arduino to display the current value of the binary pattern. The system implementation via *Fritzing* software [31] is depicted in Figure 3.

In this experiment, by the use of the Arduino as the attacker device we transferred a binary pattern to stop the normal operation of the sensor network. We tried to create a similar case to the scenario of the real application. In that case, if the sensors, which are contaminated by the HWT are deployed in an unattended area, a malicious adversary can get access to the sensors and uses an attacker device to submit a binary pattern to activate the trojan on the sensor and destroy the routine operation of the network.

*2) **Attack using PIR sensor**:* For the second experiment, we used a PIR sensor that is assembled on the Arduino board. PIR (Passive Infrared) sensor is used to detect a human target that has moved in or out of the sensor range. Every object emits some low level of radiation and the hotter something is, the more radiation is emitted. A PIR sensor is able to detect levels of infrared radiation. When a PIR detects motion in its range, its output pin will be ”high”.

As we have programmed our system, whenever the PIR sensor detects a motion in its proximity, a high signal will be generated as its output. Using the SPI bus, we transfer this signal on the MOSI line to the TelosB. System implementation is depicted in Figure 4.

On the Arduino, we also use a red LED to visualize the output pin of the PIR sensor, when the output is high, the red LED turns on. In the absence of movement around the PIR sensor, the TelosBs have their radio communication. When an activity is detected around the PIR, the communication of the sensors will be stopped, which we can easily see as the LEDs on the sensors will stop working. By disabling the SPI communication, sensors will resume their normal operation. If the attacker establishes the SPI communication with one of the sensor boards, and keeps it active, the network can be disrupted frequently based on the output of the PIR sensor.

Similar to the previous section, we can consider this experiment for the real application. The adversary can design an attacker system that will be sensitive to movement in its surrounding area. Any motion can trigger this attacker which in turn can cause the HWT on the sensor to be activated.

*3) **Attack using Morse code pattern**:* In this test, we want to create a text message to break the sensors' communication. For our purpose we use the *Morse code*. As mentioned earlier, text information can be transmitted as sequences of on-off tones, lights, or clicks. A skilled receiver can understand the message without using any tools.

Here, we design our algorithm's Morse section to receive

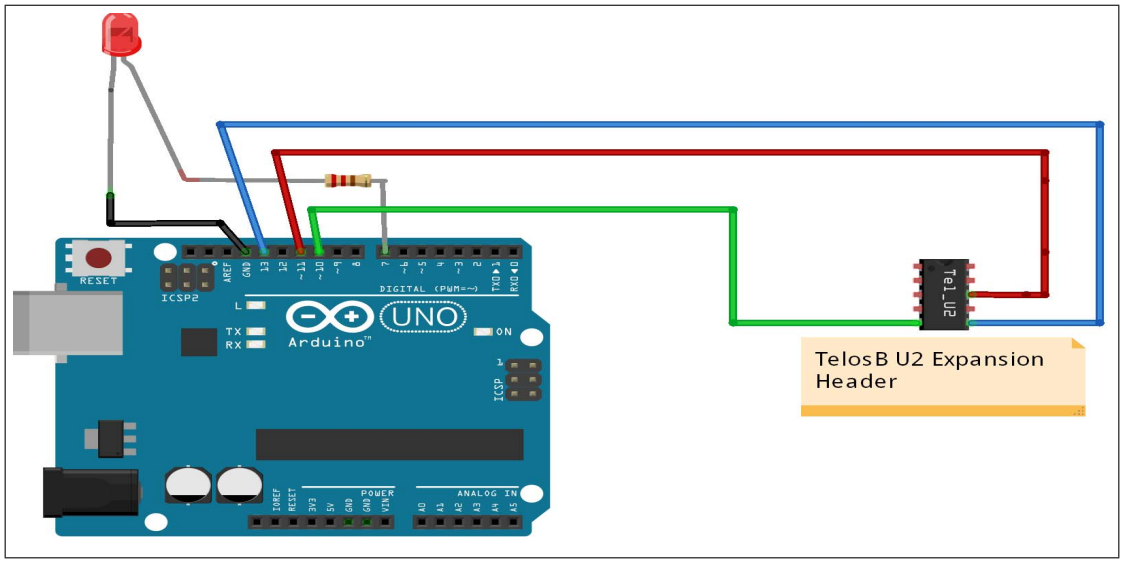

Fig. 3. Attacker using binary pattern.

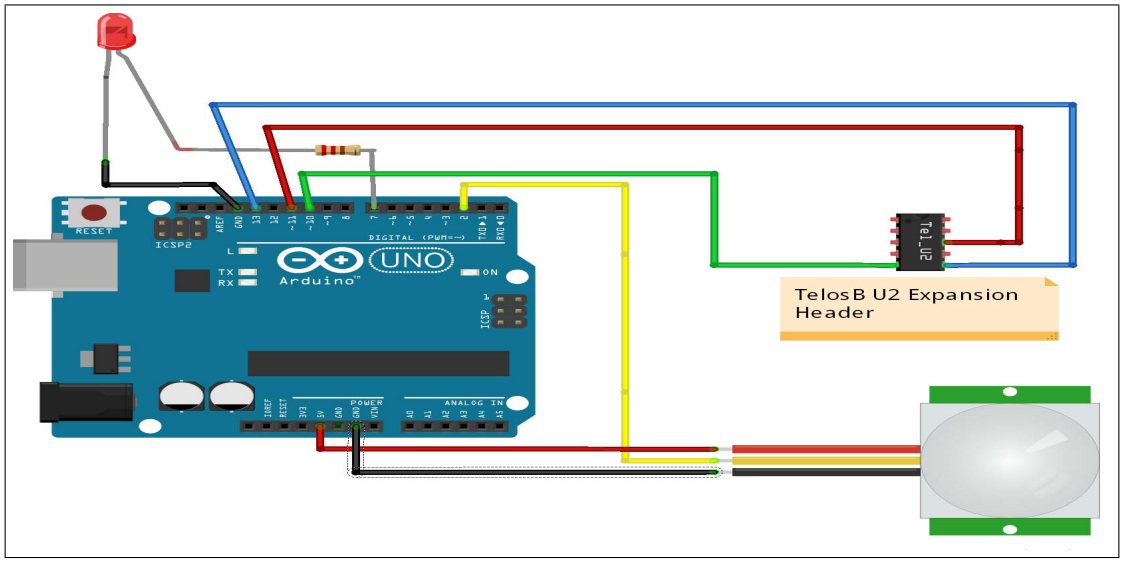

Fig. 4. System Diagram with PIR.

any form of text messages from the user via serial port of the Arduino. That is, the attacker can enter different text messages to be transmitted by SPI bus for disrupting the sensors' communication. The Morse code works as follows:

When a text message is received via the algorithm, each letter is represented using a series of *dots* and *dashes*. These two terms are used to represent short and long signals. There are a number of international rules for sending Morse code. The duration of a dash signal is three times the duration of a dot signal. Each of these signals will be followed by a short duration of silence which is equal to the time of one dot signal. The letters of a word in the text are separated using a space that is same as three dots. The words of a sentences are spaced by seven dots. The duration of one dot signal is used to represent unit of time measurement. The representation of English alphabet and numbers in Morse code is shown in Figure 5. For instance, the word trojan will be represented as:

$$\| \text{T -} \quad \text{R .-.} \quad \text{O - - -} \quad \text{J .- - -} \quad \text{A .-} \quad \text{N -.} \|$$

To make it easier to understand, we demonstrate the output of the Morse code on a LED that is attached to the Arduino. Before the attacker enters a text on the serial terminal of the Arduino, the sensors have the normal radio communication. But as soon as the attacker enters a text, it will be translated to Morse code and will be transferred via SPI communication; as the result, the sensors' communication will be stopped and LEDs on the sensors do not blink anymore. The TelosBs' communication would resume only after the physical reset button on the sensors will be pushed. The attacker can enter

A ·–	J ·–––	S ···	1 ·———–
B –···	K –·–	T –	2 ··–––
C –·–·	L ·–··	U ··–	3 ···––
D –··	M ––	V ···–	4 ····–
E ·	N –·	W ·––	5 ·····
F ··–·	O –––	X –··–	6 –····
G ––·	P ·––·	Y –·––	7 ––···
H ····	Q ––·–	Z ––··	8 –––··
I ··	R ·–·	0 –––––	9 ————·

Fig. 5. International Morse code.

any combination of text or numbers as the input to the Morse code and via the SPI bus, this can be used to interrupt the sensors' communication. In the real application of sensor network, the adversary can design the attacker device to act on a specific Morse Code. For instance, he can use *start* and *stop* as the inputs to the Morse code system to activate or deactivate the hardware trojan on the sensors.

As can be seen through the experiments that we have conducted, by exploiting the SPI bus on the sensors, it seems that a backdoor to the network is provided for the attackers which can be used the engage the sensors to act maliciously. If such a backdoor is provided for the attackers, they can implement even more complicated attacks to affect the sensors' functionality in much more severe ways or extend the number of malicious sensors to a larger amount in the network. Considering the limitations of the sensors in terms of computation resources and longevity, attackers may not need complex tools to manipulate the unattended sensors and activate the embedded hardware trojan.

V. Conclusion and Future Work

Wireless sensor networks (WSNs) have been used in many applications such as remote environmental monitoring and target tracking. They are deployed in the network for a long duration of time. In some of the applications of the WSNs, sensors are left unattended in the network. Thus, it is possible that an adversary in the network can get physical access to the sensors and makes some malicious changes.

To reduce the cost of fabricating ICs, the manufacturing process has been shifted to untrusted foundries. As a consequence, ICs are more exposed to malicious changes in the harware which is called hardware trojan and could have huge effect on their functionality. HWTs are designed to be able to escape validation test and later when the IC is in use, an adversary can find a way to activate the HWT to engage the device to act in a malicious way.

In this work, our goal was to explain that the feasibility of the HWT threat in the sensor network should be taken into consideration as sensors of a WSN are potential candidates for insertion of the HWT in the fabrication process. When the infected sensors are deployed in the field to collect data, an adversary can activate the HWT on these sensors to destroy the network.

As HWTs are extremely hard to detect and we did not have access to the special devices that are used for hardware trojan detection in our experiments, we assumed that our sensors have been contaminated with HWT and we tried to find a way to make the sensors behave in an abnormal way. For the TelosB sensors, we employed the SPI bus on the sensors and we showed that an attacker can use this weak point as a backdoor to disrupt the normal operation of the sensors which can be considered as triggering the HWT. Through our experiments, we used different methods to generate the triggering source for breaking into the TelosB.

As the result of our experiments we were able to validate that an adversary can activate a HWT on the TelosB sensors. This was one approach for investigating the threat of hardware trojans in WSNs. We believe as the sensor networks are good candidates for implementing HWTs, specific security mechanism should be designed to better identify these hardware trojans in WSNs.

We aim to raise awareness of the other researchers to the potential threat that hardware trojans can cause for the sensor networks. We believe this work can be considered as the first step towards addressing and investigating this important threat in more details.

The primary goal for our future work is to design a HWT detection and defense mechanism that can be easily implemented and integrated in the sensor. A possible solution can be done via software using a framework like Di-Sec [32] that allows monitoring of the sensing components to identify malicious activities and also facilitates the implementation of a defense technique.

References

[1] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 1–12.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Analysis of a prediction-based mobility adaptive tracking algorithm," in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. IEEE, 2005, pp. 753–760.

[3] M. Castillo-Effer, D. H. Quintela, W. Moreno, R. Jordan, and W. Westhoff, "Wireless sensor networks for flash-flood alerting," in *Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on*, vol. 1. IEEE, 2004, pp. 142–146.

[4] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 2006, pp. 102–105.

[5] K. Lorincz, D. J. Malan, T. R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: challenges and opportunities," *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 16–23, 2004.

[6] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, 2006.

[7] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 145–156.

[8] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Security and Privacy, 2005 IEEE Symposium on*. IEEE, 2005, pp. 49–63.

[9] E. Shi and A. Perrig, "Designing secure sensor networks," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 38–43, 2004.

[10] H.-N. Dai, Q. Wang, D. Li, and R. C.-W. Wong, "On eavesdropping attacks in wireless sensor networks with directional antennas," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[11] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," *Security in distributed, grid, mobile, and pervasive computing*, vol. 1, p. 367, 2007.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.

[13] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware trojan: Threats and emerging solutions," in *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International*. IEEE, 2009, pp. 166–171.

[14] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware trojan horse detection using gate-level characterization," in *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*. IEEE, 2009, pp. 688–693.

[15] F. Kiamilev, R. Hoover, R. Delvecchio, N. Waite, S. Janansky, R. McGee, C. Lange, and M. Stamat, "Demonstration of hardware trojans," *DEFCON*, vol. 16, 2008.

[16] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A novel technique for improving hardware trojan detection and reducing trojan activation time," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 112–125, 2012.

[17] Y. Jin, N. Kupp, and Y. Makris, "Experiences in hardware trojan design and implementation," in *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*. IEEE, 2009, pp. 50–57.

[18] A. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[19] E.-H. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 8. IEEE, 2006, pp. 3383–3389.

[20] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 259–268.

[21] Z. Tun and A. H. Maw, "Wormhole attack detection in wireless sensor networks," *World Academy of Science, Engineering and Technology*, vol. 46, p. 2008, 2008.

[22] X. Wang, S. Chellappan, W. Gu, W. Yu, and D. Xuan, "Search-based physical attacks in sensor networks," in *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*. IEEE, 2005, pp. 489–496.

[23] X. Wang, W. Gu, S. Chellappan, K. Schosek, and D. Xuan, "Lifetime optimization of sensor networks under physical attacks," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 5. IEEE, 2005, pp. 3295–3301.

[24] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*. IEEE, 2008, pp. 15–19.

[25] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson, "Trojan side-channels: Lightweight hardware trojans through side-channel engineering," in *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer, 2009, pp. 382–395.

[26] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware trojan detection and isolation using current integration and localized current analysis," in *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS'08. IEEE International Symposium on*. IEEE, 2008, pp. 87–95.

[27] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "Mero: A statistical approach for hardware trojan detection," in *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer, 2009, pp. 396–410.

[28] H. Salmani, M. Tehranipoor, and J. Plusquellic, "New design strategy for improving hardware trojan detection and reducing trojan activation time," in *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*. IEEE, 2009, pp. 66–73.

[29] Advanticsys, "Mtm-cm5000-msp." [Online]. Available: http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html

[30] A. DATASHEET, "8-bit avr® microcontroller with 4/8/16/32k bytes in-system programmable flash," 2010.

[31] F. Inc. [Online]. Available: http://fritzing.org/home/

[32] M. Valero, S. S. Jung, A. Uluagac, Y. Li, and R. Beyah, "Di-sec: A distributed security framework for heterogeneous wireless sensor networks," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 585–593.