Journal of Hardware and Systems Security (2018) 2:251–265
https://doi.org/10.1007/s41635-018-0040-7

Securing Medical Devices Against Hardware Trojan Attacks Through Analog-, Digital-, and Physiological-Based Signatures

Taimour Wehbe[1] · **Vincent J. Mooney**[1] · **Omer T. Inan**[1] · **David C. Keezer**[1]

Received: 1 December 2017 / Accepted: 22 May 2018 / Published online: 19 June 2018

Abstract

Corruption of data in embedded and medical devices can cause serious harm if not quickly detected. In this paper, we emphasize the part of the attack surface which entails inserting malicious hardware circuitry (Hardware Trojans) during the manufacturing process of a digital microchip. The Hardware Trojan (HT) is composed of a few gates and attempts to modify the functionality of the chip. Such types of extremely small HTs are hard to detect using other conventional offline HT detection methods, such as side-channel analysis and digital systems test techniques. In our approach, however, we focus on an online method for rapidly detecting HTs at runtime by checking for correct functionality of the underlying hardware. We present an architecture that addresses these threats by splitting the design into a two-chip approach where we generate signatures deep in the hardware during data harvesting, and we then check for these signatures during data processing and encryption for transmission. In addition, we take advantage of known physiological relationships between medical data to ensure the integrity of the data that is processed by the hardware. Our experimental results demonstrate the effectiveness of our HT detection architecture and show that not only can we detect such types of attacks but also that we can distinguish these attacks from actual health problems. Our synthesis results show that our architecture minimally impacts performance and area especially in light of the fact that most of our techniques rely on digital logic modules which are already typically present in modern digital chips for test and other purposes.

Keywords Hardware trojans · Hardware security · Embedded reconfigurable logic · Medical devices · Physiological feature extraction · Ballistocardiography · Electrocardiography · Digital systems test · Multiple Input Signature Register (MISR) · Analog signatures

1 Introduction

The disaggregation of the chip manufacturing industry has increased the threat of malicious insertion of undesirable logic functions into fabricated digital chips. Such logic functions have been shown to have the ability to leak sensitive information or even alter the functionality of a system. The aforementioned stealthy hardware modifications are referred to in the literature as Hardware Trojans (HTs). The effects of HTs can be disastrous if an attack targets sensitive applications such as medical or embedded military devices. For example, in 2010, the US Navy discovered missiles provisioned with fake microchips with a "back door" that could have been used to remotely shut the missiles down at any time, rendering them useless [1]. Over the past decade, research agencies and groups have been looking for ways to verify that digital chips are not hacked during the production process. For instance, in 2016, the IEEE Center for Secure Design and the Food and Drug Administration (FDA) released reports spotlighting security red flags for the wearable industry, one of which is falsifying a user's health data by physically manipulating the device [2, 3].

Designing techniques to detect HT attacks in such small devices has proven to be a non-trivial task. Techniques that try to perform offline testing, such as side-channel analysis

✉ Taimour Wehbe
taimour.wehbe@gatech.edu

Vincent J. Mooney
mooney@ece.gatech.edu

Omer T. Inan
omer.inan@ece.gatech.edu

David C. Keezer
david.keezer@ece.gatech.edu

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA

or digital systems test, have so far not been able to guarantee the security of the chip. In addition, online techniques that try to detect HT attacks at runtime need careful design considerations due to the energy consumption and computing power limitations for these devices.

In this paper, we present an embedded systems HT detection architecture that combines multiple HT detection techniques [4–6]. We target HTs that are extremely small in size and which, once triggered, attempt to modify the functionality of the chip by attacking the user's data. Our work is motivated by a health monitoring scenario which includes sensors capturing heart signals and transmitting them for further processing and analysis [7]. The physiological signals have known relationships which we take advantage of to create multiple types of signatures that check for the integrity of the captured data at runtime. Specifically, we embed different techniques of signature generation deep in the hardware (during data harvesting), and then we check for the validity of these signatures during digital processing to ensure that the chip has no HT attacks and that the data's integrity is maintained.

Our work improves consumer confidence in health monitoring applications by increasing the assurance provided to the public that their captured health data are correct. For example, consider a scenario where health data collected by such applications are used to interact with drug titration procedures. An attack using HTs can cause misleading analysis of the patient's health leading to possible lawsuits.

2 Background and Prior Work

2.1 Hardware Trojan Taxonomy and Detection

Over the past decade, a more formal HT taxonomy has been introduced [8]. Figure 1 shows how different types of HT attacks can be classified according to three broad characteristics: (i) physical, (ii) activation, and (iii) action characteristics.

Quite a few HT detection methods [4–6, 8–17] have been proposed to address specific types of the aforementioned attacks. These methods can be broadly divided into side-channel analysis, HT triggering, and correct functional verification techniques.

1) Side-channel analysis: This type of HT detection relies on analyzing information gained from the physical implementation of an architecture and comparing the information to data generated by the normal behavior of a chip to detect anomalies. Energy consumption, timing analysis, and electromagnetic emanations are the most common techniques for side-channel analysis [8–10]. However, these types of methods lack the ability to detect HTs that are small in size due to the minor effects that the HT might cause in terms of power and timing variations [11, 12].

2) Hardware Trojan triggering: This type of HT detection architecture relies on extensively testing the design prior to deployment and use (i.e., right after chip fabrication). Prior work has proposed methods to detect HTs with significant hardware footprints, e.g., using a "golden die" [13]. The authors of [14, 15] provide a survey of multiple types of HT detection methods including methods that are based on HT activation mechanisms. HT detection approaches based on a "golden die" present a variety of disadvantages. First, cleverly inserted HTs may not be easily triggered by these approaches as the testing mechanisms have no idea of the presence of the HT and/or its location in the chip. Thus, the HT might pass the testing phase undetected. Second, this type of detection does not guarantee the correctness of the design at runtime. Finally, performing an offline full-functionality test for each fabricated chip might turn out to be inefficient and time-consuming, thus increasing the cost of microchips. Our proposed architecture addresses all of these shortcomings by checking for the data correctness while trying to detect HT attacks at runtime. In addition, our technique is HT size independent and

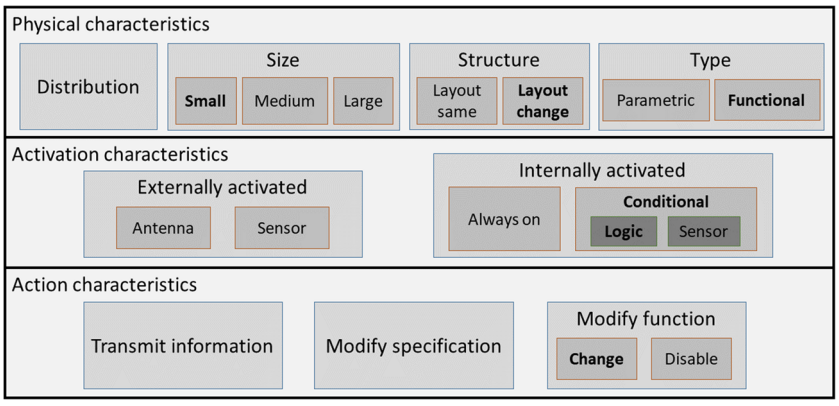

Fig. 1 A Hardware Trojan taxonomy characterizing HTs according to their physical, activation, and action properties

can catch ultra small HTs since we look for the effects generated by an HT momentarily after it is activated.

3) *Functional verification*: This type of HT detection relies on checking the functionality of the hardware by monitoring the output and checking for expected behavior. The authors of [8, 14] provide a survey of multiple types of HT detection methods including methods that are based on architecture-level detection and functional verification. Others have devised techniques that solely rely on functional verification for checking the trustworthiness of the underlying hardware [16, 17]. Our work falls under this category where we create hardware signatures to check for the correct functional behavior of the digital microchip at runtime [4–6].

2.2 Digital Hardware Signature Generation

In this work, we rely on the technologies provided in the literature to compute a "signature" or a hash of a bitstream where the signature is compact yet has a very low probability of generating an identical signature for two different bitstreams under the threat scenario that we consider (see Section 3). Therefore, if an HT attempts to modify either the signature or the input data stream, the mismatch can be identified with a high level of certainty. In our previous work [4, 16], we made use of digital systems test techniques to create hardware signatures. Those types of signatures were close in nature to hash-based signatures where a signature was generated from one sensor data input. Specifically, we used a form of a Linear-Feedback Shift Register (LFSR) called a Built-in Logic Block Observer (BILBO) Multiple-Input Signature Register (MISR), which is normally used in digital systems test to compress multiple sets of data and generate a signature [18]. Figure 2 shows a 4-bit BILBO register that can be configured as a MISR for signature generation. The major incentive behind our use of MISRs is their advantage in terms of area and energy consumption [4, 16].

2.3 Electro- and Ballistocardiography

To convey the effectiveness of our architecture, we consider the health monitoring scenario shown in Fig. 3. The figure shows a person standing on a scale and holding on to a handlebar. Both sets of aparatus are used as a part of a system that measures the person's vitals. In our work, we are interested in electrocardiogram (ECG) and ballistocardiogram (BCG) sensors that are normally used to monitor and capture heart pulse electrical and timing events. In Fig. 3, the handle bar is used as a grip-style dry electrode for ECG measurement [19]. The BCG is a measurement of the movements of the body in response to the ejection of blood by the heart and the movement of blood through the vasculature [7, 20]. As shown in Fig. 3, the BCG scale could also be equipped with postural sway adjustment pumps. The person's cellphone (shown in the figure) could be used to capture 3D on-body inertial measures to help adjust the person's postural sway. This helps in generating improved and more accurate ECG and BCG readings. In that case, the BCG hardware would be running a real-time application with data processing to provide the necessary balancing mechanisms. Harvested ECG and BCG data are typically further processed and analyzed to extract physiological features that assist in the diagnosis of health conditions [7, 21].

The ECG and BCG sensors that we use in our experiments have analog output values that fall within a range of -0.9999 to 0.9999. In addition, the health monitoring application requires an accuracy of four significant digits after the decimal. In some cases in our architecture, values have to be squared and added, and in which case, the range of 0 to 1.9999 needs to be supported. Thus, to cover the range and provide the needed accuracy, we use a signed 16-bit fixed-point format with the most significant bit as the sign bit, the next bit as a representation of a value of 1 or 0, and the remaining 14 bits representing the fractional part of the number.

3 Threat Scenario

One of the most interesting classes of HT attacks is the type that stealthily targets the functionality of a specific digital design via minimal logic insertion. This type of attack typically relies on a trigger condition as shown in Fig. 1. In this paper, we consider the threat model shown in

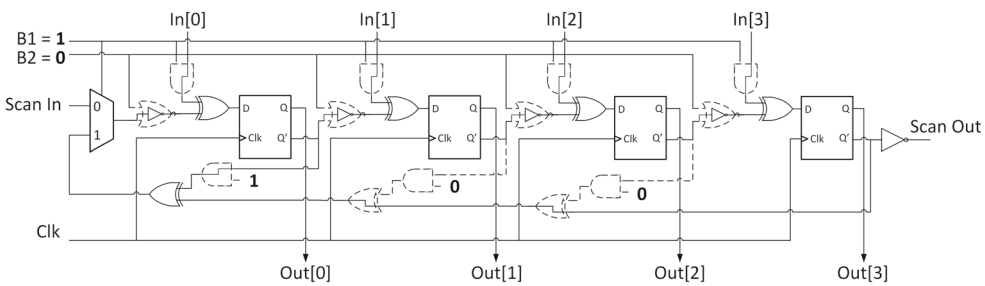

Fig. 2 A digital systems built-in self test BILBO module configured to operate as a MISR

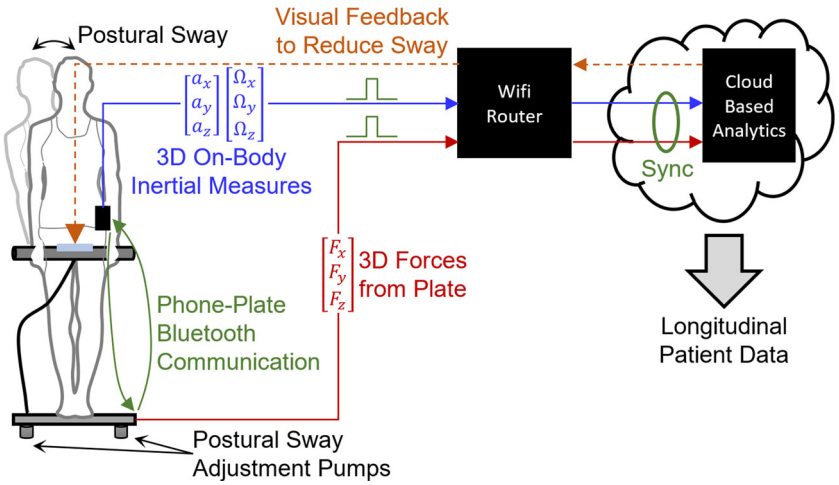

Fig. 3 A medical device example showing a person standing on a BCG scale and holding onto an ECG handlebar to capture the heart signals in real time. The BCG scale could also be equipped with postural sway adjustment pumps to improve the accuracy of sensor data

Fig. 4. The HT model, which is representative of most prior work [11–14, 16], is composed of *trigger circuitry* and a *payload*.

As shown in Fig. 4, the *payload* can be as small as a single exclusive-or (XOR) gate that modifies data on a bus in the design (*Input* bus signal in this case) by toggling a single bit (most significant bit in this case). As Fig. 4 shows, such a type of HT attack is extremely small (just a few gates) and cannot be reliably detected if spread out among thousands or even millions of gates in an embedded system design [12]. In addition, the effect of such an attack could be disastrous if it goes undetected for relatively long periods of time. For instance, in the example scenario shown in Fig. 3 and discussed in Section 2.3, an attack on the ECG or BCG data could hypothetically cause the adjustment pumps on such a BCG scale to behave abnormally in a way that could harm the person standing on the scale.

Figure 5 shows an example of trigger circuitry responsible for waiting for an activation characteristic to trigger the HT. In our threat scenario, we consider HT trigger circuitry which is based on a conditionally triggered HT. As Fig. 5 shows, the trigger circuitry can be designed using only a *counter* and minimal *control* logic. The *counter* is attached to a *rarely toggling node* in the *processing block* on the chip. The *counter* is incremented every time the value on the wire toggles. The *control* logic monitors the output of the counter and asserts the *trigger* once the *counter* reaches a specific predefined value.

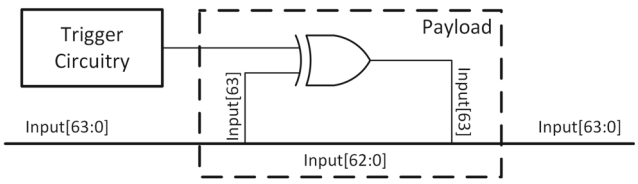

Fig. 4 Our HT threat model showing trigger circuitry and a payload composed of a simple XOR gate

It is important to note that in our threat model, we assume that once the HT is triggered, it remains on for a relatively long time. That is because if an attacker wants to intelligently turn on the HT for finite periods to bypass specific detection techniques, the trigger circuitry (shown in Fig. 5) will have to be more complex resulting in an HT with a larger size. Such types of larger sized HTs are beyond the scope of our work and, as indicated in Section 2.1, can be caught by other HT detection techniques [8–10, 14, 15].

4 Hardware Trojan Detection Architecture

To target the specific types of HT attacks presented in our threat scenario in Section 3, we devise an architectural level solution where we split our design into two separate chips. Our approach is similar in principal to the concepts presented in [22, 23] where a prover and a verifier are used together to verify correct execution on a chip. Figure 6 shows an overview of our proposed architecture. Our method relies on the idea of generating different types of signatures during data harvesting by sensors and then checking for these signatures later during data processing and encryption.

The left chip in Fig. 6 represents our first chip or "Chip 1." This chip is responsible for performing analog-to-digital conversion and initial signature generation. Thus, we assume that Chip 1 can use older technologies and can be fabricated in a trusted fab where, for example, employees

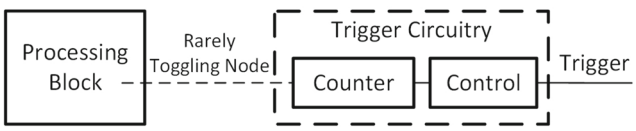

Fig. 5 An HT trigger circuitry example composed of a rarely toggling node that increments a counter to set the HT trigger

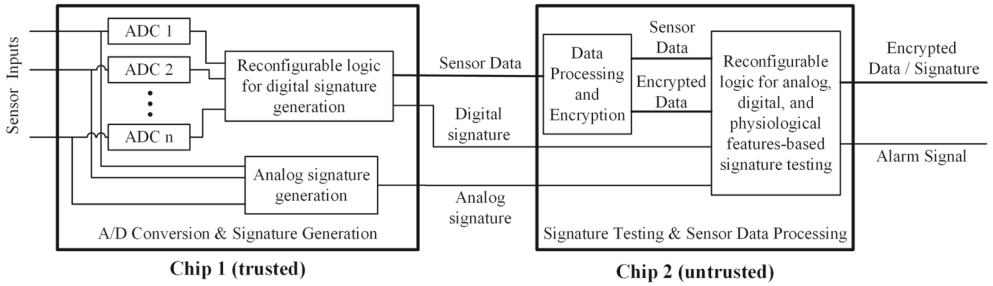

Fig. 6 An overview of our HT detection architecture showing how a design can be split into two chips. Chip 1 performs signature generation during data harvesting, and Chip 2 checks for these signatures during processing and prior to transmission

have security clearances. In our design, we choose to create two types of signatures, a digital signature and an analog-based signature. The two created signatures in Chip 1 along with the captured data are then passed on to Chip 2 in Fig. 6. Chip 2 is responsible for data processing, data encryption, and signature regeneration and testing. Specifically, in this work, we perform up to three types of signature checks: digital-, analog-, and physiological-based signature testing. Alarm signals are generated out of each of these signature testing mechanisms. The alarm signals are used to inform upper level firmware or software of any potential health problem or hardware attack/error. Chip 2 is assumed to be fabricated using a state-of-the-art process node, usually by another company in a different country, to provide the required complexity and performance needs of such a chip. Thus, we consider that an HT can be injected into any part of this chip including the primary inputs. To detect such types of HT attacks, including ones that target the signature testing logic, it was reasonable for us to embed our signature generation and testing mechanisms in reconfigurable logic providing the fabrication company with no information regarding how to insert HTs and allowing for the flexibility in terms of tailoring the testing mechanisms to the application's security needs over the course of the chip's lifetime.

Incorporating signature generation and testing deep in the hardware provides several advantages. First, it enables checking the integrity of the data as early as possible. Second, it improves the security of the overall embedded system design by further complicating the job of a potential malicious entity. Finally, having multiple signature generation techniques–i.e., digital-, analog-, and physiological-based signatures–helps in complicating the job of an attacker and in improving the analysis and decision making of our architecture. In fact, simultaneously issuing multiple alarm signals by three independent types of signature generation techniques reduces the false positives and more importantly the false negatives of our approach.

4.1 Digital Signature Generation and Testing

4.1.1 Chip 1

Figure 7 shows a more detailed block diagram of the part responsible for generating the digital signature from the sensor data in Chip 1. In this work, we use the BCG force that is along the vertical direction, i.e., the BCG Head-to-Foot (HF) signal, along with the ECG signal. ECG and BCG HF sensor inputs are first passed through analog to digital (A/D) converters before being momentarily stored into First-In-First-Out (FIFO) buffers. The FIFO buffers are 16 bits wide and two slots deep giving the ability for every FIFO to store two consecutive sets of sensor inputs. The output of every FIFO is concatenated to form 32-bit blocks which are in turn concatenated to form a single 64-bit block. The resulting 64-bit block is fed into a MISR [4, 16]. The MISR receives 64 consecutive 64-bit blocks and compresses them into a single 64-bit signature (shown as *Signature 1* in Fig. 7). *Signature 1* along with the sensor data are passed on to Chip 2.

4.1.2 Chip 2

Figure 8 presents a closer look at the components present in Chip 2. First, the digital sensor data coming from Chip 1 are concatenated to form a 64-bit block. Second, every block is encrypted using an encryption cipher, such as PRESENT [24], to generate the ciphertext (*CT* in Fig. 8). Third, the encrypted data is passed through a decryption cipher to

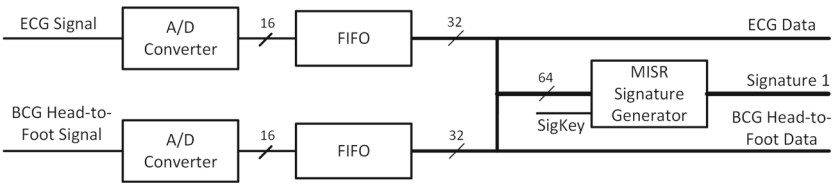

Fig. 7 A closer look at the logic design that generates the digital (MISR) signature in the first chip

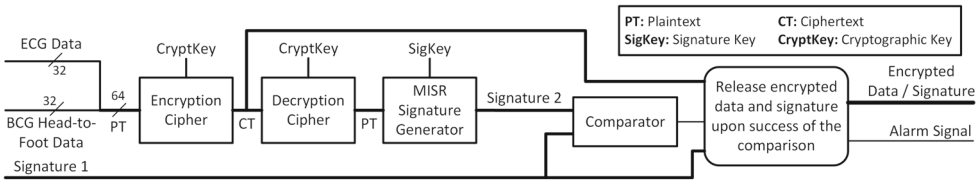

Fig. 8 A closer look at the logic design that checks for the correct digital (MISR) signature in the second chip

regenerate the sensor data plaintext (*PT* in Fig. 8). *PT* regeneration helps in detecting HT attacks and hardware errors in the encryption and decryption ciphers [4, 16]. Namely, if the *PT* is directly used for signature regeneration, an HT attack on the encryption or decryption cipher will not affect the regenerated signature and thus the attack goes undetected. Fourth, the *PT* is fed through the MISR digital signature generator to regenerate the signature (*Signature 2* in Fig. 8). Finally, *Signature 1*, the signature coming in from Chip 1, is compared to *Signature 2* to check for any HT attacks or hardware errors in Chip 2. The result of the comparison is fed to a release logic block (right-hand side of Fig. 8) which is responsible for releasing the encrypted data upon success of the comparison. The alarm signal indicates the presence of a potential HT attack or error.

It is important to note that for the signatures that we consider (MISR signatures), we are not aware of known techniques (under the considered threat scenario of extremely small HTs) to quickly compute how a change to an input bit would affect the associated signature. Specifically, since we embed HT detection (MISR signature generation and testing) in reconfigurable logic, it appears infeasible for an HT to simultaneously change both the input and the signature in a way that would avoid detection, especially given that the primitive polynomial of the BILBO MISR (Fig. 2) is not yet placed in the chip hardware logic at fabrication time. Therefore, the MISR signature is accepted to have properties that do not fully satisfy ones of a cryptographic signature. However, if the application requires higher security guarantees, a designer could opt to use a much more secure signature generation algorithm such as a version of the Secure Hash Algorithm at the expense of area and energy consumption. In addition, in this work, we do not address key storage (*CryptKey* and *SigKey* in Fig. 8); however, effective techniques such as utilizing Physically Unclonable Functions (PUFs) for generating keys on the fly can be used [25].

4.2 Analog Signature Generation and Testing

4.2.1 Chip 1

Figure 9 shows a detailed view of the architecture responsible for the generation of the analog signature in Chip 1 in Fig. 6. First, ECG and BCG HF data harvested from sensors are amplified and filtered to improve the signal-to-noise ratio of the captured data. Second, the filtered data is fed to analog-to-digital converters and prepared to be passed to Chip 2 for encryption and further processing. At the same time, the harvested sensor data are passed through an analog signature generation logic.

In our approach, we create an analog signature using the vector sum (i.e., square root of the sum of the squares of the ECG and BCG measurements) of the captured data as shown in the bottom right-hand side of Fig. 9. One of the major reasons behind using the vector sum as an analog signature is its importance in analyzing and diagnosing health problems in heart monitoring medical devices [26, 27]. Therefore, we take advantage of the need for such calculations to improve the security by using it to check for the integrity of the captured sensor data. In addition, the simplicity and the abundance of the major components (adders, squarers, and square root modules) [28] needed to perform such a calculation make it an intriguing candidate for analog signature generation in embedded devices.

Finally, the created analog signature is fed into an analog-to-digital converter and passed on to Chip 2 along with ECG and BCG sensor data.

4.2.2 Chip 2

Figure 10 shows a detailed view of the architecture responsible for the digital signature generation and comparison that happens in Chip 2 in Fig. 6. Similar to the process described

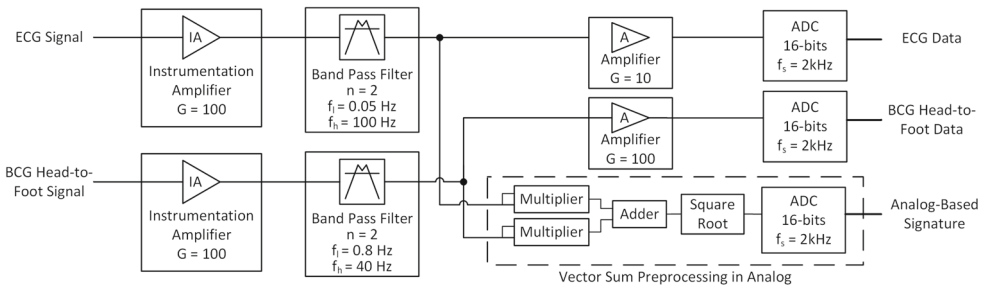

Fig. 9 A closer look at the analog signature (vector sum) generation in the first chip

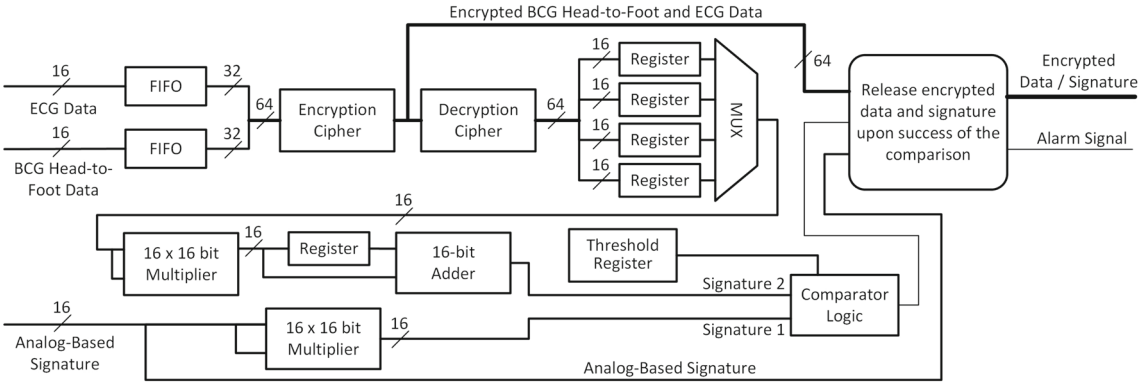

Fig. 10 A closer look at the hardware of the second chip which is responsible for checking if the analog-based signature matches the digital vector sum calculation

in Section 4.1.2, the ECG and BCG data coming in from Chip 1 are concatenated to form 64-bit blocks that are in turn encrypted and decrypted using the PRESENT encryption and decryption ciphers respectively. However, the decrypted data in this case are split back into four 16-bit registers. Each two registers hold two consecutive sets of ECG and BCG data. The control unit in Chip 2 schedules the passage of each set of data through the multipliers and adders to regenerate a digital version of the vector sum. Specifically, the first ECG data set is passed through the 16-bit multiplier where the data is squared and saved in a register. Then, the first BCG data set is passed through the multiplier to be squared and then added to the squared ECG data using a 16-bit adder. The generated result is denoted as *Signature 2* in Fig. 10. It is to be noted that since the analog chip is sampling the BCG data using 16-bit analog-to-digital converters, the 32-bit result of the multiplier is truncated and the most significant 16 bits of the result are used as inputs to the next stage. The loss of precision in generating the analog-based signature using 16-bit multipliers directly affects the application at hand. As mentioned in Section 2.3, in our case, the health monitoring application requires an accuracy of only four significant digits after the decimal. Thus, the need for 14 bits to represent the fractional part of the value is enough for this application.

Synchronously, the analog-based signature coming from Chip 1 is passed through a 16-bit multiplier acting as a squarer to generate the squared value of the vector sum (sum of squares of the ECG and BCG data). The result of the squarer is referred to as *Signature 1* in Fig. 10. *Signature 1* and *Signature 2* are then compared to determine if an HT attack or hardware error has happened anywhere in the second chip (Chip 2). Similar to what is shown in Fig. 8, the result of the comparison is fed into a release logic block to determine whether the encrypted data can be safely transferred out of the chip. If the comparison fails, an alarm signal is raised indicating a potential HT attack or error.

It is to be noted that the comparator logic for the analog-based signature testing, shown in Fig. 10, behaves in a slightly different fashion than the one used with digital-based signature testing, shown in Fig. 8. In this case, the comparator logic declares a match if $|Signature1 - Signature2| \leq threshold$; otherwise, a signature mismatch is declared. Note that $threshold$ is an input set by the user due to the analog nature of the application and the signature. To reduce the rate of false positives introduced by the analog-based vector sum signature comparison, the threshold register is typically monitored during a learning period for every individual and is adjusted according to the person's data. The threshold is however limited by the accuracy needed by the application at hand. For example, if the application cannot tolerate any difference between the analog-based signature and the regenerated signature, the false positive rate would drastically increase due to the architecture flagging a possible HT attack when in fact the difference is due to the rounding of the analog signature. Thus, for this signature generation technique, a compromise has to be made between relaxing the threshold and detecting attacks targeting the least significant bits of the signature.

4.3 Physiological Features-Based Signature Generation and Testing

As opposed to the digital- and analog-based signature generation and testing techniques, the physiological-based signature generation and testing technique does not require a signature generation phase in Chip 1 in Fig. 6. Instead, this technique extracts features (signatures) only in Chip 2 and relies on the known physiological properties of these features to check for anomalies in the circuit that affect the integrity of the data.

Figure 11 shows a detailed view of the architecture responsible for generating and testing for the physiological-based signatures in Chip 2 of our design. As with the

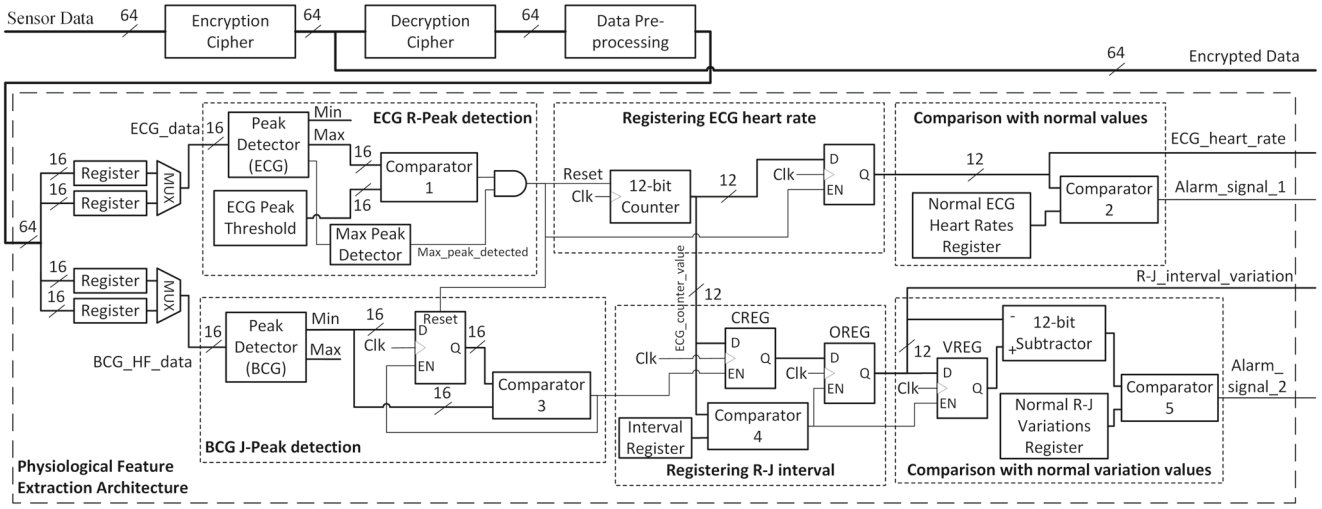

Fig. 11 Hardware implementing two physiological features extraction and testing. Namely, the heart rate and the R-J interval of an individual are being computed in real time and compared to known normal values

previous two techniques, sensor data (ECG and BCG HF data) coming into Chip 2 are first encrypted and decrypted using the encryption/decryption cipher. Then, the decrypted data is pre-processed to improve its signal-to-noise ratio for a better feature extraction. The concatenated sets of ECG and BCG data are split back into sets of 16-bit signals. The ECG and BCG data sets are then used to extract two main physiological features, namely, the heart rate and the R-J interval [7, 21, 26, 29].

The heart rate feature is calculated by using a peak detection architecture (design by Jordanov and Hall [30]) to find the ECG peaks (R-peaks) in consecutive ECG data sets. The time difference between the ECG R-peaks is registered and is defined as the heart rate. Once the heart rate is calculated, it is compared to a threshold of normal heart rate values to detect abnormalities.

The R-J interval is calculated by finding the time difference between an ECG R-peak and the global maximum (J-peak) in the first 400 ms of the BCG HF signal [29]. Our architecture looks for all the maxima in the BCG HF signal using the same peak detection architecture and registers only the J-peak. Once the J-peak is detected, the R-J interval is found by calculating the time difference between the ECG R-peak and the BCG J-peak. The variation of the consecutively calculated R-J intervals is then checked against normal threshold values to detect abnormalities.

As shown in Fig. 11, the physiological feature-based signature generation and testing architecture generates two alarm signals, namely, *Alarm_signal_1* and *Alarm_signal_2*. The nature of these two alarm signals differs from the ones generated by both the digital-based signature testing architecture and the analog-based one. The physiological-based alarm signals indicate one of three status values: "no anomaly," "anomaly" with high fidelity, and possibility of an anomaly which is referred to in our work as the "gray zone" [5]. Determining the severity of the alarm signals depends on the different ranges of the extracted features based on physiology and as reported in the literature [29, 31].

5 Combining Digital-, Analog- and Physiological-Based Signatures

One of the important aspects of our design is its complimentary nature where it can be applied in parallel with other HT detection techniques. In this section, we show that the techniques introduced in Section 4 can be all combined together into one architecture to not only detect HT attacks and hardware errors in medical devices, but also distinguish them from health problems. Similarly, other types of signature generation techniques can be seamlessly integrated into our architecture if needed.

The modified block diagrams of Chip 1 and Chip 2 of our overall design are shown in Figs. 12 and 13 respectively. Figure 12 shows that Chip 1 has to be slightly modified to combine both the analog and digital initial signature generation. Specifically, FIFO buffers are introduced to store multiple sets of data. The stored sets of data are then passed through a BILBO MISR block to create the MISR-based signature (digital signature) which is passed along with the analog-based signature (vector sum of the ECG and BCG HF data) to Chip 2.

Figure 13 shows the detailed architecture of Chip 2. First, the ECG and BCG HF data are encrypted and decrypted;

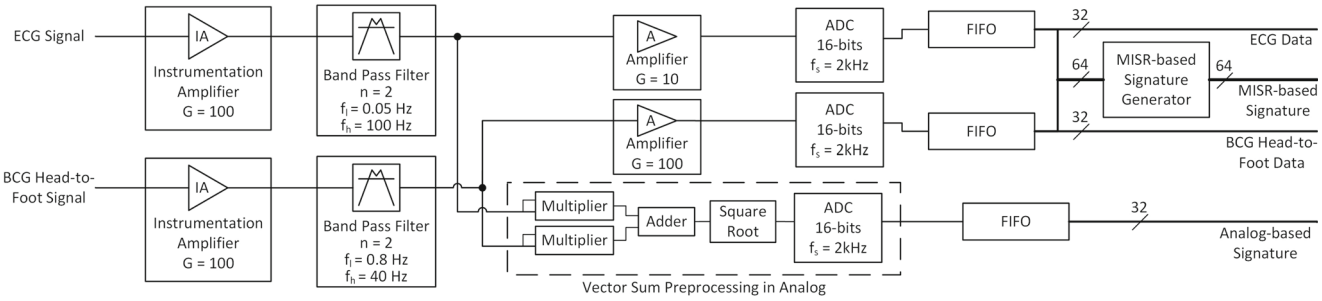

Fig. 12 Final implementation of the first chip (Chip 1) showing the generation of both digital and analog signatures during the data harvesting process

then, the resulting plaintext is used to generate all three types of signatures, the digital-, analog-, and physiological-based ones. Thus, the release logic is now controlled by multiple alarm signals coming from two types of signature comparators (*Comparator Logic 1* and *Comparator Logic 2* in Fig. 13) and from the heart rate and R-J interval feature extraction circuitry.

It is important to note that adding multiple types of testing for the integrity of data is important for specific types of HT attacks and increases the security level of the architecture by allowing for smarter decisions when analyzing the results of multiple signature generation techniques. In addition, the incorporation of signature comparison (for both digital- and analog-based signatures) along with physiological feature extraction helps in distinguishing HT attacks or other hardware errors from health problems.

To better show the analysis of the different alarm signals of Fig. 13, we present in Table 1 some plausible decisions that could be taken upon looking at the combination of all alarm signals in the architecture. For example, an HT attack-modifying low order bits of the BCG HF data and the analog-based signature might go undetected in the architecture shown in Fig. 10 (see Section 6.2). However, for an architecture where even detection of such changes to low order bits is needed, adding the MISR-based HT detection approach may be desirable, especially if the digital systems test structures already include BILBO logic which can be reconfigured for use in HT detection. Adding the MISR-based HT detection would help in detecting an attack on any of the data bits as the authors are unaware of any publication that shows a way of simultaneously modifying the data bits and the MISR-based signature without being caught [4, 16]. As shown in Table 1, even when the analog-based signature module indicates a match, if both the digital and physiological-based signature modules indicate a mismatch, the architecture can then guarantee that an anomaly is present in the system.

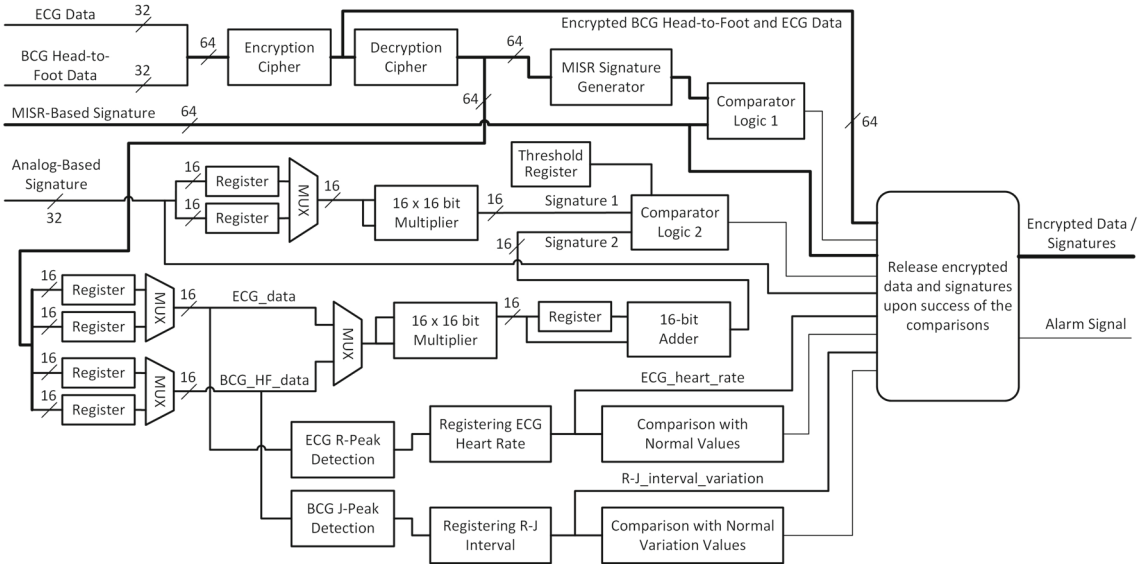

Fig. 13 Final implementation of the second chip (Chip 2) showing the regeneration and testing of digital-, analog-, and physiological features-based signatures

Table 1 Analysis of combining digital-, analog-, and physiological-based signature generation and testing

		Digital-based Signature			
		Match		***Mismatch***	
		Analog-based Signature		***Analog-based Signature***	
		Match	***Mismatch***	***Match***	***Mismatch***
Physiological-based Signature	***No Anomaly***	Correct operation	Possible attack on the analog signature generation architecture	Possible HT attack or hardware error in low order bits	Possible occurrence of a false negative by the physiological-based signature architecture
	Gray Zone	Potential health problem (person is recommended to seek medical help)	Potential health problem or possible attack on the analog signature generation architecture	Possible HT attack or hardware error along with a potentially minor health problem	Hardware Trojan attack or hardware error
	Anomaly	Person is asked to seek immediate medical help	Possibility of a serious health problem or attack on the analog signature generation architecture	HT attack or hardware error along with a possibility of a serious health problem	Hardware Trojan attack or hardware error

Another example that benefits from the incorporation of the analog-, digital-, and physiological-based approaches is the case of an HT attack on the ECG and BCG HF data inputs in the architecture of Fig. 13. Suppose the HT were to consist of some minimal logic to swap the values of the ECG and BCG HF signals. This type of specific attack will go undetected by the sole use of the analog-based signature detection approach. However, the attack will be detected and flagged by the combined architecture as shown in Table 1.

Finally, as Table 1 shows, the combination of the three techniques not only improves the decision making in whether an anomaly is present in the circuit, but also helps in identifying whether the anomaly is due to an actual hardware attack/error or due to a health problem that the individual is having. For example, if both the analog- and digital-based techniques indicate a match while the physiological-based feature extraction circuitry indicates an anomaly, the person is asked to seek immediate medical help indicating a high possibility of a serious health problem.

It is important to note that in this work, our architecture focuses only on creating means of detecting HT attacks, hardware errors, and/or health problems by asserting alarm signals when the respective conditions are believed to be present. The decisions and countermeasures are subject to higher level policies and protocols.

In addition, since our techniques rely on the correctness of signature generation and comparison, it is worth mentioning the chip aging effects on the reliability of our approach. The analog-based vector sum and the physiological-based signature generation architectures already implement threshold registers due to the inherent variations in the values of these signatures. These threshold register values can be calibrated as the chip ages to compensate for signature changes over time. However, the MISR-based signature generation presents a more challenging scenario since single bit flips would generate false positives. Techniques used to address the aging effects in hardware implementations of hashing algorithms and PUFs, such as the use of error detection and correction, could be added to our architecture to compensate for chip aging [32].

6 Hardware Trojan Attack and Detection

To test our architecture against different types of HT attacks, we studied four different classes, two of which are attacks that target a single point in the architecture (referred to as "single attacks"), and the other two of which are attacks that target multiple points (referred to as "coordinated attacks"). The attacks we aim to address are numbered according to their type from 1 to 4.

6.1 Single Attacks

HT attack types 1 and 2 fall under the class of single attacks. Both types of attack attempt to modify the data in the same way as presented in our threat scenario described in Section 3 and shown in Figs. 4 and 5. Attack type number 1 is an attack on the input data as soon as it appears on Chip 2 (Fig. 13). This attack type is probably the most important and our work [4–6] appears to be the first to provide an architecture able to detect such type of an attack. The difficulty in detecting this type of attack is that it happens before even any encryption or signature generation has been performed. Other types of signature-based HT

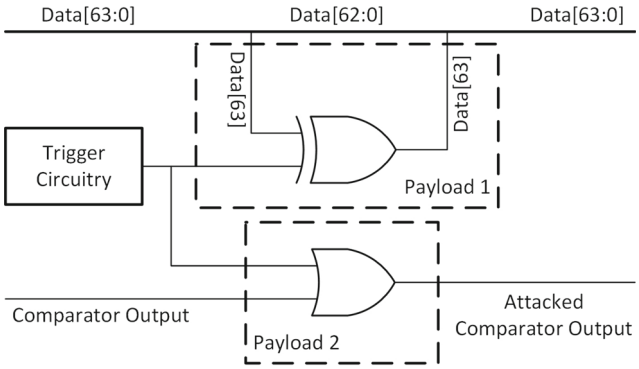

Fig. 14 An HT attacking both an internal data bus in the design and the output of a comparator

detection techniques fail to detect this attack as their signature generation scheme relies on the input data [13, 16]. However, in our detection approach, since the initial signature is generated in Chip 1 (Fig. 12), the attack at the input of Chip 2 (Fig. 13) can be caught.

Attack type number 2 targets data at the outputs of internal modules in the architecture. For example, an HT might try to flip a single bit at the output of the encryption or decryption cipher. This results in the generation of an altered plaintext which results in the creation of altered digital-, analog-, and physiological-based signatures, thus triggering the multiple alarm signals in our architecture. However, this type of attack has been well studied in the literature [13, 16, 17].

6.2 Coordinated Attacks

HT attacks of types 3 and 4 fall under the class of coordinated attacks. Attack type 3 is devised in a way where the HT trigger circuitry is connected to two payloads. The first payload attacks the data attempting to flip a single bit either at the input of the chip or at the output of any of the internal modules as discussed in attack types 1 and 2. The second payload attacks the output of the comparator logic forcing the comparator to always indicate a success (Fig. 14). This way, even if the data was altered resulting in an altered regenerated signature, the comparator will still indicate a success in the comparison and the attack will go undetected.

To prevent such a type of attack, for every comparator in the design, we insert a module in reconfigurable logic, the *comparator testing logic* block (Fig. 15), to specifically check for the attack's effect on the comparator. The comparator testing logic periodically checks for the expected behavior of the comparator by asserting a *Test Mode* signal as shown in Fig. 15. In this way, the regenerated signature is intentionally modified and the result of the comparison is checked. If the *comparator* indicates a match, then we know with a high fidelity that the *comparator* is under attack. When the *Test Mode* signal is deasserted, the regenerated signature is passed without any alteration and the circuit behaves normally. It is worth mentioning that for energy constrained embedded and medical systems, such as battery-powered devices, extensive testing might not be feasible and thus the comparator testing logic might be simplified or completely removed. An alternative could be to directly embed the comparators in reconfigurable logic to strengthen the architecture against HT attacks that attempt to provide a workaround against our proposed signature checking technique.

Attack type 4 presents a case where the HT tries to modify a low order bit in an input signal (ECG or BCG HF signal) along with a low order bit in the analog-based signature. The possibility of the success of this attack is due to the analog-to-digital conversion nature where the comparator in our architecture (*Comparator Logic 2* in Fig. 13) allows for a specific threshold of difference between the analog-based signature (coming from Chip 1) and the regenerated signature in Chip 2. However, the digital-based signature (MISR) and the physiological-based signature comparisons will detect the attack as shown in Table 1. In addition, it is unclear what ability the attacker would gain by changing the low order bits as these slight variations in the values of the inputs and the signature may also occur due to the analog nature of the application.

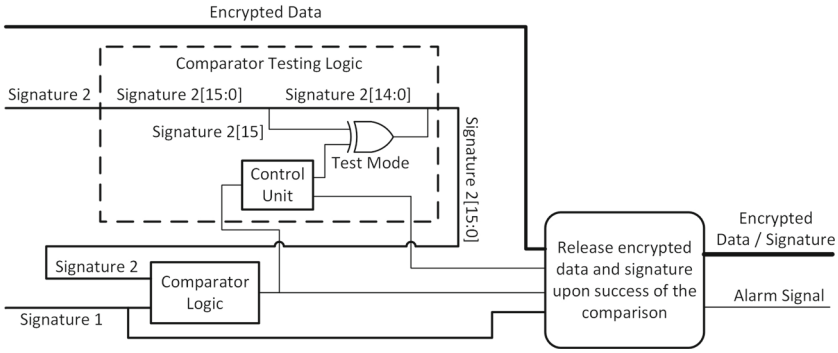

Fig. 15 A comparator testing logic unit inserted to verify the correct operation of a comparator and detect any HT attacks that attempt to alter the comparator's result

7 Experimental Results

The digital components of our architecture were implemented using VHDL code, simulated using Mentor Graphics ModelSim version 10.6*a* and synthesized using Synopsys Design Compiler version J-2014.09. We tested our architecture against multiple types of HT attacks on the ECG and/or BCG HF data that were captured over 60 s and sampled at a 2-kHz rate from six different individuals. The subjects were healthy and at rest during the capture process. The human subjects measurements were approved by the Georgia Tech Institutional Review Board, and subjects provided written informed consent. When running the experiments, three different user health conditions that correspond to healthy individuals, individuals with minor heart problems, and individuals with severe heart conditions were simulated.

7.1 Simulation Results and Functional Verification

We simulated multiple HT attacks of all four types presented in Section 6. In our simulations, we set the *Threshold Register* of *Comparator Logic 2* in Fig. 13 to a hexadecimal value of "0008" which represents a value of 2^{-11} in our fixed-point representation. In addition, we set the ECG heart rate alarm ranges to represent "anomaly" when the values are below 30 bpm (beats per minute) and above 150 bpm; "no anomaly" when the values are between 45 and 110 bpm, and "gray zone" otherwise [31]. Normal R-J interval variation values were set to represent "anomaly" when the variation is above 50%; "no anomaly" when the variation is below 15% and "gray zone" for variations inbetween [29]. Moreover, our HT trigger circuitry was configured to monitor for a specific occurrence of a BCG HF data sample, e.g., a hexadecimal value of "12*CF*" which represents approximately 0.29388 mV in our fixed-point representation. Once that same value has appeared for 64 times, the HT was launched.

7.1.1 Simulation of Single Attacks

To simulate HT attack types 1 and 2, we inserted HT logic similar to the one shown in Fig. 4 with some cases targeting input data as it arrives on chip and others targeting data inside the chip. For example, in some of our simulations, we inserted HT logic that attacks the ECG input data as soon as it appears on the chip shown in Fig. 13. This resulted in modifying the encrypted and decrypted data which in turn lead to the modification of the MISR-based signature along with the regenerated analog-based signature (*Signature 2*) right before signature comparison. Once the maliciously modified signatures were compared to the original signatures coming from Chip 1 (Fig. 12), *Comparator Logic 1* and *Comparator Logic 2* both declared mismatches. In addition, the physiological feature extraction circuitry generated abnormal ECG heart rates and R-J interval variations confirming the mismatches of the signatures. Therefore, the release logic prevented the transmission of the data and asserted the alarm signal. Moreover, other simulations of the same attack type, this time at the output of the encryption cipher, confirmed the need to decrypt the data and recreate the signature from the regenerated plaintext rather than directly from the input.

7.1.2 Simulation of Coordinated Attacks

When simulating attack type 3, we implemented two HT payloads affecting two different points in the architecture as shown in Fig. 14. *Payload 1* attacks the output of the encryption cipher eventually leading to modifications in the regenerated signatures. Simultaneously, *Payload 2* forces the output of the comparator to show a match even when the compared signatures do not match. It is important to note here that the comparator testing logic is periodically checking for this specific case. In our simulations, the periodicity was set to 16 iterations, i.e., *Test Mode* in Fig. 15 is set to 1 after 16 sets of data have been processed through the architecture. *Test Mode* is asserted for only one clock cycle where the system is stalled and the comparator output is checked for legitimate operation. Therefore, the release logic might transmit altered encrypted data depending on when the HT is triggered. However, performing the testing periodically can solve the problem if the sets of data between two consecutive tests (in our case, 16 sets) can be declared invalid if attack type 3 was detected (a multi-bit alarm signal can encode different types of alarm conditions, e.g., a specific bit encoding of the alarm could be used to indicate failure of the comparator testing logic).

To simulate attack type 4, the HT was designed to wait for the same triggering mechanism and then attack the low order bits of both the BCG HF input data and the analog-based signature shown in Fig. 13. Our simulations show that such type of attack results in modifications to the values of *Signature 1* and *Signature 2*; however, these modifications are minimal (below the threshold of *Comparator Logic 2*) and are not detected by the analog-based signature testing mechanism. Fortunately, the MISR signature generation and testing method detects these types of attacks as any single bit flip in the original input to the signature generator generally results in multiple bit flips in the generated signature and therefore is detected by *Comparator Logic 1*. In addition and as expected, the physiological features (ECG heart rate and R-J interval variation) indicated minor alarm severity (gray zone) since the modifications in the features were minimal and not conclusive by themselves. However, when coupled with the mismatch indicated by the MISR

comparator logic, the release logic was able to confirm the possibility of an HT attack and assert the chip's overall alarm signal.

7.1.3 Timing and Efficiency of Attack Detection

To compare the behavior of the different signature generation and testing techniques described in Sections 4.1 through 4.3, we reran the same types of HT attacks while varying the bit location which the HT inverts from the most significant bit (MSB) to the least significant bit (LSB). The aggregated results of the average time taken to detect an HT for each of the three techniques are presented in Fig. 16. It is important to note that for the physiological features-based technique, an HT is considered to be detected if the alarm signal status indicates an anomaly or a gray zone condition. This does not apply to the remaining two techniques (MISR-based and analog-based) as these two techniques have only two alarm severity levels, a signature match and a signature mismatch status.

The experimental simulation results showed that the analog-based signature technique was the fastest in detecting HTs, i.e., within a few clock cycles, but the least accurate. The digital-based signature (MISR) technique was the most accurate and was able to detect all types of HTs. However, MISR-based signatures took longer time to declare an alarm after an HT was triggered. That is because they require compressing multiple sets of data and then checking for the integrity of the whole set. The accuracy of the physiological-based signature technique relied on the type of the HT attack. For example, for HTs that attack low order bits, a small portion were not reported (less than 5%) by the method. In addition, attacks on low order bits of the input data took longer time to detect, while attacks on the most significant bits were detected at a much faster rate. Thus, the physiological-based signature technique proved to provide moderate HT detection accuracy, but was the slowest in HT detection primarily due to the fact that the technique inherently requires multiple sets of data to construct physiological features.

As expected, the simulation results showed that combining all three signature-based techniques into one architecture provided the ability to successfully detect all the different types of HT attacks defined in the threat model in the fastest time possible (within few clock cycles). In addition, the combination of the physiological-based signatures with both the analog and digital-based signatures not only allowed for the detection of HT attacks and hardware errors in medical devices, but also helped in distinguishing them from health problems as reported earlier in Table 1.

7.2 Synthesis Results

The digital modules of our combined architecture were synthesized using the Synopsys Design Compiler version J-2014.09 for Linux and were mapped to the *NCSU* 45-nm Base Kit Library [33]. Table 2 shows the area results of the main modules of our design post synthesis. It is obvious that a significant area of the architecture is covered by the encryption/decryption and processing modules. The security modules that are inserted to regenerate and test for the integrity of the data consume, as expected, a significantly lower area.

To better show the area overhead imposed by introducing our HT detection technique, we compute the overall area usage of the digital chip containing only the processing hardware and encryption/decryption units and compare it to the overall area of our modified architecture which includes the HT detection circuitry. The results show that the overhead introduced by the physiological-based signatures mechanism is the lowest (around 4%) assuming that the physiological features are part of the processing that is done on chip. The analog-based signature generation and

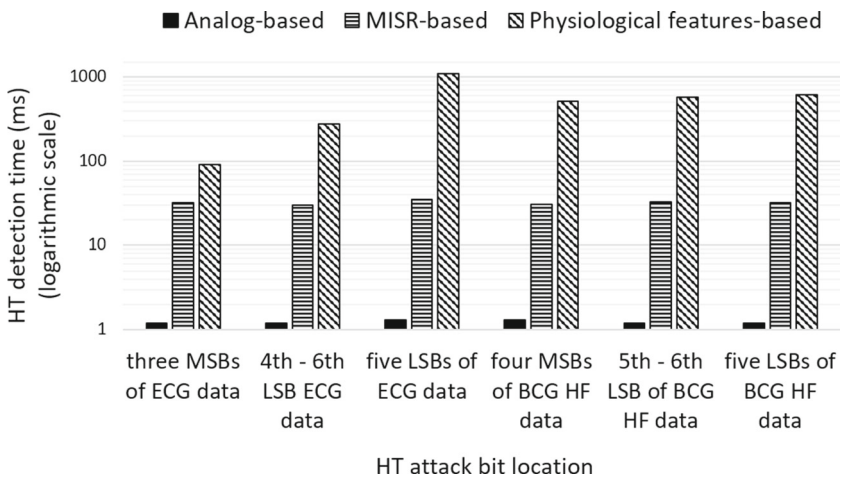

Fig. 16 Time taken by each of the signature testing techniques to detect HTs targeting different data bit locations

Table 2 Area results of the major digital components of our HT detection architecture

Module	Area (square microns)	Area (kGE)
Encryption Cipher (PRESENT [24])	5517	2.939
Decryption Cipher (PRESENT [24])	5431	2.893
MISR-based Signatures Overhead	6172	3.288
MISR-based Signatures Overhead (shared with digital systems test)	3575	1.904
Analog-based Signatures Overhead	1839	0.98
Physiological Feature Extraction	3485	1.856
Physiological-based Sig. Overhead	954	0.508
Release Logic	2414	1.286

testing mechanism came in second with an overhead of around 7% while the MISR-based signature generation and testing technique had an overhead of around 14%. It is to be noted that our architecture allows for the use of any of the proposed techniques by themselves or any combination depending on the needed security of the application at hand and the available resource and power limitations. Clearly, a combination of the three proposed signature generation and testing methods achieves the highest confidence in detecting and distinguishing HT attacks and hardware errors from health problems at the expense of a higher area and power consumption.

It is also important to note that in our experiments, Chip 2 contained only encryption and decryption blocks. In more realistic scenarios, such a chip could contain other processing and transmission modules which require larger area. Our conclusion is that the percentage overheads reported earlier can be considered pessimistic as increasing the overall chip area would eventually decrease the overhead of our HT detection approach.

Our current design achieves a maximum clock frequency of 300 MHz. An analysis of the timing results show that the multiplier that is used in the generation of the analog-based signature (*Signature 2* in Fig. 13) falls along the critical path of our architecture. We currently implement the squaring operations in our design using Synopsys DesignWare's combinational carry save array multiplier. As reported by Synopsys [34], this type of implementation has a delay of 3.25 ns. If the application requires a higher clock speed a designer can choose to map the multiplier's logic to other implementations. For example, DesignWare has a Booth-recoded Wallace-tree multiplier which has a delay of 1.6 ns (for a 16-bit multiplier). In addition, DesignWare provides other options of pipelined and sequential multipliers. Choosing between these types of implementations allows the designer to make area versus delay tradeoffs.

8 Conclusion

In this work, we present a novel two-chip architecture for detecting malicious hardware modifications at run-time in medical devices through the use of hardware signatures. Specifically, our architecture detects extremely small HTs in embedded medical devices which when triggered attempt to modify the functionality of the design. Three different techniques for signature generation were developed, namely, analog-, digital-, and physiological-based signatures, by taking advantage of known relationships between health sensor data.

Our functional simulation results showed that the developed architecture successfully detects the targeted types of HTs defined in the threat model including ones that target single points and multiple points in the architecture. Our synthesis results show that it is feasible to implement our HT detection architecture with minimal area overhead. In addition, timing analysis shows that our implemented HT detection circuitry does not fall on the critical path of the design and thus does not affect its speed.

Compliance with Ethical Standards The human subjects measurements were approved by the Georgia Tech Institutional Review Board, and subjects provided written informed consent.

References

1. Johnson R The Navy bought fake Chinese microchips that could have disarmed U.S. missiles, Business Insider, July 2011. [Online]. Available: http://www.businessinsider.com/navy-chinese-microchips-weapons-could-have-been-shut-off-2011-6
2. West J, Kohno T, Lindsay D, Sechman J (2016) WearFit: Security design analysis of a wearable fitness tracker, IEEE Center for Secure Design
3. Post-market management of cybersecurity in medical devices, Center for Devices and Radiological Health, Food and Drug Administration, U.S. Department of Health and Human Services and Center for Biologics Evaluation and Research, 2016
4. Wehbe T, Mooney V, Keezer D, Parham NB (2015) A novel approach to detect hardware Trojan attacks on primary data inputs. In: Proceedings of the 10th Workshop on Embedded Systems Security (WESS), pp 2:1–2:10
5. Wehbe T, Mooney VJ, Javaid AQ, Inan OT (2017) A novel physiological features-assisted architecture for rapidly distinguishing health problems from hardware Trojan attacks and errors in medical devices. In: 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp 106–109
6. Wehbe T, Mooney VJ, Keezer DC, Inan OT, Javaid AQ (2017) Use of analog signatures for hardware Trojan detection. In: Proceedings of the 14th FPGAworld Conference
7. Inan OT, Migeotte PF, Park KS, Etemadi M, Tavakolian K, Casanella R, Zanetti J, Tank J, Funtova I, Prisk GK, Rienzo MD (2015) Ballistocardiography and seismocardiography: a review of recent advances. IEEE J Biom Health Inf 19(4):1414–1427
8. Tehranipoor M, Koushanfar F (2010) A survey of hardware Trojan taxonomy and detection. IEEE Des Test Comput 27(1):10–25

9. Bhunia S, Hsiao MS, Banga M, Narasimhan S (2014) Hardware Trojan attacks: threat analysis and countermeasures. Proc IEEE 102(8):1229–1247
10. Lamech C, Plusquellic J (2012) Trojan detection based on delay variations measured using a high-precision, low-overhead embedded test structure. In: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp 75–82
11. Wei S, Potkonjak M (2013) The undetectable and unprovable hardware Trojan horse. In: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pp 1–2
12. Wei S, Li K, Koushanfar F, Potkonjak M (2012) Hardware Trojan horse benchmark via optimal creation and placement of malicious circuitry. In: DAC Design Automation Conference, vol 2012, pp 90–95
13. Wu TF, Ganesan K, Hu YA, Wong HSP, Wong S, Mitra S (2016) TPAD: Hardware Trojan prevention and detection for trusted integrated circuits. IEEE Trans Comput Aided Des Integr Circuits Syst 35(4):521–534
14. Francq J, Frick F (2015) Introduction to hardware Trojan detection methods. In: 2015 Design, Automation Test in Europe Conference Exhibition (DATE), pp 770–775
15. Moein S, Subramnian J, Gulliver TA, Gebali F, El-Kharashi MW (2015) Classification of hardware Trojan detection techniques, in 10th Int'l Conf. on Computer Engineering Systems (ICCES), pp 357–362
16. Gbade-Alabi A, Keezer D, Mooney V, Poschmann AY, Stöttinger M., Divekar K (2014) A signature based architecture for Trojan detection. In: Proceedings of the 9th Workshop on Embedded Systems Security (WESS), pp 3:1–3:10
17. Sullivan D, Biggers J, Zhu G, Zhang S, Jin Y (2014) FIGHT-metric: Functional identification of gate-level hardware trustworthiness. In: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pp 1–4
18. Abramovici M, Breuer M, Friedman A (1990) Digital systems testing and testable design. IEEE Press, Piscataway
19. Richard E, Chan ADC (2010) Design of a gel-less two-electrode ECG monitor. In: Int'l Workshop on Medical Measurements and Applications, pp 92–96
20. Prisk GK, Verhaeghe S, Padeken D, Hamacher H, Paiva M (2001) Three-dimensional ballistocardiography and respiratory motion in sustained microgravity. Aviat Space Environ Med 72:1067–1074
21. Etemadi M, Inan OT, Giovangrandi L, Kovacs GTA (2011) Rapid assessment of cardiac contractility on a home bathroom scale. IEEE Trans Inf Technol Biomed 15(6):864–869
22. Wahby RS, Howald M, Garg S, Shelat A, Walfish M Verifiable ASICs. In: 2016 IEEE Symposium on Security and Privacy (SP), pp 759–778, vol 2016
23. Imeson F, Emtenan A, Garg S, Tripunitara MV (2013) Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation. In: Proc. of the 22nd USENIX Conf. on Security (SEC), USENIX Association, pp 495–510
24. Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, Seurin Y, Vikkelsoe C (2007) PRESENT: an ultra-lightweight block cipher. Springer, pp 450–466
25. Maes R, Van Herrewege A, Verbauwhede I (2012) PUFKY: A fully functional PUF-based cryptographic key generator. In: Prouff E, Schaumont P (eds) Cryptographic hardware and embedded systems – CHES 2012. Springer, Berlin, pp 302–319
26. Javaid AQ, Fesmire NF, Weitnauer MA, Inan OT (2015) Towards robust estimation of systolic time intervals using head-to-foot and dorso-ventral components of sternal acceleration signals. In: 2015 IEEE 12th international conference on wearable and implantable body sensor networks (BSN), pp 1–5
27. Ashouri H, Inan OT (2016) Improving the accuracy of proximal timing detection from ballistocardiogram signals using a high bandwidth force plate. In: 2016 IEEE-EMBS international conference on biomedical and health informatics (BHI), pp 553–556
28. Coughlin RF, Villanucci RS (1990) Introductory operational amplifiers and linear ICs: theory and experimentation. Harlow. Pearson Education Limited, United Kingdom
29. Wiens AD, Etemadi M, Roy S, Klein L, Inan OT (2015) Toward continuous, noninvasive assessment of ventricular function and hemodynamics: Wearable ballistocardiography. IEEE J Biom Health Inf 19(4):1435–1442
30. Jordanov VT, Hall DL (2002) Digital peak detector with noise threshold. In: IEEE Nuclear Science Symp. Conf. Record, vol 1, pp 140–142
31. Schlant RC, Adolph R, DiMarco J, Dreifus L, Dunn M, Fisch C et al (1992) Guidelines for electrocardiography. A report of the American college of cardiology/American Heart Association Task Force on assessment of diagnostic and therapeutic cardiovascular procedures. J Am Coll Cardiol 19(3):473–481
32. Yu MD, Devadas S (2010) Secure and robust error correction for physical unclonable functions. IEEE Des Test Comput 27(1):48–65
33. NCSU 45nm FreePDK^{TM} process design kit. Electronic Design Automation, North Carolina State University. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK
34. Syed AH Performance of different multipliers in the DesignWare building block IP, DesignWare Technical Bulletin, Synopsys Inc