Design of a Piezoelectric Based Physically Unclonable Function for IoT Security

Carson Labrado, Himanshu Thapliyal
University of Kentucky, Lexington, KY USA

***Abstract*—According to a report from McAfee and the Center for Strategic and International Studies, worldwide financial loss due to cybercrime was estimated to be $600 billion in 2017. Researchers are currently exploring new methods for preventing cybercrime in IoT devices. Physically Unclonable Functions (PUFs) show promise as a device that could help in the fight against cybercrime. PUFs are a class of circuit that are unique and unclonable due to inherent variations caused by the device manufacturing process. We can take advantage of these PUF properties by using the outputs of PUFs to generate secret keys or pseudonyms that are similarly unique and unclonable. In recent years, energy harvesting devices such as piezoelectric devices have been integrated with IoT devices for various purposes such as power generation and sensing applications. In this work we propose a PUF design based on piezo sensors which are already commonly found in IoT devices. Our proposed PUF is tested in terms of reliability and uniformity.**

I. Introduction

SOME researchers believe that the Internet of Things (IoT) will be the main component of the next era in computing [1]. IoT is a network of smart devices that are connected via the internet. A smart device can be described as an internet-enabled embedded system. The Internet of Things is not limited to only simple devices such as sensors and actuators. Instead, IoT includes a wide selection of systems with varying complexities such as home appliances, mobile devices, vehicles, etc. Through IoT, a myriad of connected devices are able to exchange information as components of intelligent applications.

Unfortunately, the rise of IoT has also coincided with a rise in cybercrime as information transmitted between IoT enabled devices can be the target of cyberattacks. A recent report from McAfee and the Center for Strategic and International Studies (CSIS) has estimated that in 2017 the total global loss due to cybercrime was close to $600 billion, nearly 1% of the world's GDP [2]. This is a sharp increase from their 2014 report that estimated worldwide losses to be $445 billion [3]. As a response to this growing threat, many interested parties including researchers, the military, and businesses have begun to place added emphasis on introducing security measures into IoT systems to help safeguard them from cyberattacks [4] [5] [6]. One area of research that has drawn attention as a potential cybercrime countermeasure for IoT devices is Physically Unclonable Functions (PUFs). A PUF is a device that uses inherent variations caused by the manufacturing process to create unique and unclonable IDs.

In recent years, IoT devices have begun to be integrated with multiple types of energy harvesting devices including solar cells, thermoelectrics, and piezoelectric devices. These devices were integrated so that they could be used in various applications such as power generation and sensing. The performance of these energy harvesting devices can vary between individual devices. This variance can be traced back to intrinsic variations in the devices that are caused by the manufacturing process. The authors believe that these variations make energy harvesters a good candidate for PUF creation. Designing a PUF from components that are common to IoT devices should have the added benefit of simplifying the PUF's future integration with these technologies as the required underlying hardware should already be present. As a proof of concept, we have designed a PUF based on piezo sensors that can generate 128-bit responses. We tested our design in terms of reliability and uniformity to assess the viability of using piezo sensors as a means for creating a PUF.

A. Motivation

The purpose behind the creation of our PUF was to create a PUF that could be constructed from existing components. PUFs are commonly designed at the transistor level which puts some special constraints on the manufacture of each individual PUF. Those constraints are not present if the PUF is instead manufactured from existing components. Additionally, we felt out PUF would be attractive to IoT applications if it was made from components that are already common in IoT devices such as energy harvesters and microcontrollers.

There are multiple sources of energy (such as kinetic, solar radiation, thermal energy, etc.) that could be targeted by energy harvesters in the embedded devices that would be found in IoT applications [7]. For the purposes of this work we have chosen to focus on piezo sensors which use the piezoelectric effect to convert the kinetic energy contained in vibrations and other motions into electricity. The integration of this type of energy harvester has drawn interest in IoT [8] [9].

B. Related Work

1) Silicon Based PUFs: Numerous PUF designed have already been presented in the literature. Existing PUFs have generally been silicon based. Those PUFs rely on transistor level variations that occur during the manufacturing process. These variations manifest themselves as non-uniform delays between gates in each instance of the PUF. The number of variations is large enough that individual chips can be

uniquely identified despite having identical designs and being produced by an identical manufacturing process. Common designs include arbiter [10] [11], ring oscillator [12], SRAM [13], butterfly [13], latch [14], and flip-flop [15].

- Arbiter PUF - This type of PUF compares the delay of what should be two identical circuit paths. The paths compared are actually a multiplexer chain. The actual path is determined by the input challenge. Each bit of the challenge is fed to multiplexer inputs at each stage. The drawbacks to these types of PUFs are they can be susceptible to modeling attacks and difficulties in actual implementation. Since the routing paths must be completely symmetric, mapping the design in an FPGA can result in unbalanced paths.
- Ring Oscillator (RO) PUF - The RO PUF is another dealy based PUF. It compares the number of oscillations in two ring oscillators. Each ring oscillator will have differences in delays which will result in a different number of oscillations. Just like with the arbiter PUF, the need for symmetric circuit layouts make it hard to map these types of PUFs on FPGAs.
- SRAM PUF - The Static RAM (SRAM) PUF consists of a large number of memory units which are a pair of cross coupled inverters. Intrinsic variations in the gates result in each memory unit having a default value of 0 or 1 when power is first supplied to it. The PUF response is constructed from these readings. SRAM PUFs are not always suitable to FPGA mapping due to their need for a very specific layout and hardware composition. Additionally, the use of volatile memory means power must be supplied or else the response will be lost.
- Butterfly PUF - The butterfly PUF is an improved version of an SRAM PUF. It uses latches instead of inverters in its memory units. This helps alleviate some of the SRAM PUF concerns of being able to map it on FPGAs.
- Latch PUF - The latch PUF is another type of memory PUF. It uses cross coupled NOR gates for its memory units. Its reliance on a specific type of logic gate makes it difficult to map on FPGAs.
- Flip-flop PUF - The flip-flop PUF is a memory PUF that is specifically designed for FPGA implementation. It uses flip-flops that are already present on an FPGA in the same way that an SRAM PUF uses its memory units to generate a response.

2) Non-Silicon Based PUFs: Other researchers have proposed designs that are based on much larger components such as sensors, rather than transistor level designs. These designs are based on a wide range of devices including microelectromechanical systems (MEMS) based sensors [16] [17], device touchscreens [18], photodiodes [19], and solar cells [20]. The MEMs designs [16] [17] utilize a MEMs gyroscope to generate responses. The touchscreen based PUF [18] requires a user to trace a specified pattern with their finger. The responses are built from the variation in how different uses try to trace the same patter. The photodiode based PUF [19] compares the voltage outputs of identical groupings of photodiodes. A conventional PUF is used to determine which sensors are being compared to generate each response bit. The solar cell work [20] shows that solar cells have intrinsic variations that could be used as a PUF, but does not actually present a complete design.

3) Relative Merits of PUF Designs: The major advantage sensor based PUFs have over silicon based ones becomes apparent during their design process. These PUFs can be readily tested and modified since they are made from existing components. The need to actually manufacture silicon based PUF designs makes the creation of physical copies prohibitive in some cases. Those designs tend instead be simulated and tested using software rather than the actual physical device. The downside to non-silicon PUF designs is the size of their components make them tend to be much larger than silicon based PUF designs.

Our proposed PUF is much closer to this later group of sensor based PUF designs as it is based on piezo sensors and microcontrollers.

C. Contribution of Work

In this paper we propose a PUF design that is specifically targeted for use in IoT applications. Our proposed design is constructed from components that are common in IoT devices such as microcontrollers and energy harvesters. We provide the following:

- Proposal of a PUF circuit design methodology that leverages intrinsic piezo sensor characteristics.
- Proposal of a balancing algorithm to generate a 128-bit response from a collection of eight piezo sensors.
- Testing the reliability of the proposed PUF over a time period of ten days.
- Testing the uniformity of the proposed PUF over a time period of ten days.
- Testing the reliability of the proposed PUF over a temperature range of -20°C to 0°C and 25°C to 80°C.

This paper is organized as follows: Section II provides more detail on PUFs and how they may be incorporated into security applications; Section III describes that methodology that was used to design our proposed PUF; Section IV explains the testing metrics of reliability and uniformity and the results of those tests; Section V compares our proposed design to existing sensor based PUF designs; and lastly, Section VI concludes the paper by describing directions for future work.

II. Background

This section will provide general information on Physically Unclonable Functions (PUFs) and describe some of their uses.

A. Physically Unclonable Functions

First proposed in [21], a Physically Unclonable Function (PUF) can be thought of as a type of hash function in which a given input will result in a specific output. In PUFs, inputs are known as ”challenges” and outputs are ”responses”. Collectively, a challenge and its associated response are known as challenge-response pair (CRP). The reason these functions are called ”physically unclonable” is because PUFs are designed

in such a way where it is impossible to create multiple PUFs that will be have identical outputs for all possible inputs. This is because PUFs use the minor variations inherent to device manufacture to generate their uniqueness property. For example, in a circuit based PUF these physical variations can include qualities such as signal propagation delay times present in wires and logic gates. A set of ideal PUFs with identical designs given identical challenges should have a unique response for each individual PUF. This uniqueness property is illustrated in Figure 1.

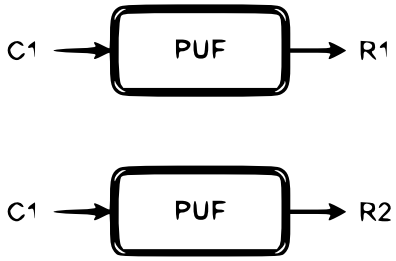

Fig. 1. Example of Uniqueness Property of PUF

In the Figure, each PUF is given a challenge denoted as $C1$. The response of the first PUF is $R1$ while the response of the second PUF is $R2$. As depicted in Figure 1, different CRPs have been produced despite providing what should be identical PUFs with identical challenges. A PUF that is considered strong will have a large number of viable CRPs while a weak PUF will have a very limited number of possible CRPs.

B. Use of PUF as a Security Measure

The inherent properties of uniqueness and unclonable of PUFs make them attractive for use in security applications. For example, an attacker would be forced to obtain the actual PUF itself if he or she wanted to use it in an attack as it would be impossible to create an exact copy of the desired PUF. Researchers have already proposed various methods for integrating PUFs into standard security applications. Perhaps the most obvious security applications of PUFs are the secure generation and storage of secret keys [22] [23]. A PUF response by itself likely can't be used as a secret key due to the reliability issues inherent in PUFs and the mathematical constraints placed on secret keys by their respective cryptosystems. PUF responses can however be used as a seed during the process of creating a secret key. In standard cryptographic systems once a secret key is created it must then be stored in secure memory to provide additional security from unauthorized access. Secure memory has the disadvantage of being more expensive and slower to access than normal unsecure memory. By using a PUF response to generate the secret key, the need to use secure memory is effectively removed. The reason for this is because the secret key never actually has to be stored by the system. Instead, the PUF can generate the secret key every time it needs to be used. The secret key is derived from the response of the PUF which means the only data that will actually need to be stored is the challenge associated with the response used to derive the desired the secret key. Due to the uniqueness of individual PUFs, a challenge by itself is essentially useless without also having access to the PUF. For this reason, the challenge can be stored in simple unsecure memory and as a result the secret key no longer requires the use of secure memory.

III. Design Methodology

In this section we will explain the design of our proposed PUF. The complete architecture of the proposed PUF consists of a microcontroller, eight piezo sensors, eight 100 KΩ resistors, and an AC voltage source. The proposed PUF should be considered a weak PUF as it is designed to have only one possible challenge-response pair. There reason there should only be one pair is because the response generated by the PUF is a result of comparing intrinsic characteristics of the piezo sensors. Because those intrinsic characteristics will not change, comparisons of those characteristics and the response derived from them should not change either.

A. Piezo Sensor

As described in [20], a piezo sensor can be modeled by the Butterworth-van-Dyke equivalent circuit shown in Figure 2. Capacitor C_0 represents the electrical capacitance between the piezo sensor leads. Capacitor C_1 represents the mechanically equivalent capacitance inversely proportional to the stiffness of the piezo sensor. Inductor L represents the mechanically equivalent inductance proportional to the mass of the piezo sensor. Finally, Resistor R represents the losses across the piezo sensor.

The presence of capacitors and inductors in the equivalent circuit guarantees that the equivalent impedance of the piezo sensor can be varied by connecting an AC voltage source to the leads of the sensor and and varying its frequency. In theory, multiple copies of the same model of piezo sensor should have identical parameters each component in their equivalent circuit. In actuality, the manufacturing process introduces slight variations into individual sensors. These variations result in individual sensors having unique characteristics which manifest as the component values in the equivalent circuit. The uniqueness of individual sensors can be utilized to create a PUF.

B. Basic Piezo Circuit Diagram

The circuit shown in Figure 3 forms the basic building block of our proposed PUF. The circuit consists of an AC voltage source V_S, a piezo sensor, and a 100 KΩ resistor R placed in series. Assuming the peak amplitude of the input voltage source remains constant, then the voltage V_R across resistor R will be determined by the impedance Z_Piezo of the piezo sensor which in turn will be determined by the frequency of the voltage source. Across multiple copies of this circuit, even when they all have identical input voltage sources, the voltage V_R across resistor R will not actually be consistent. These voltages will instead vary due to the previously described unique intrinsic characteristics of each piezo sensor.

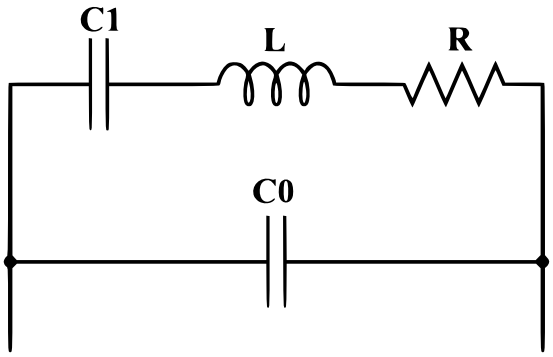

Fig. 2. Piezo Sensor Butterworth-van-Dyke Equivalent Circuit

These unique intrinsic characteristics will manifest as unique impedance values of Z_Piezo in each circuit and as a result V_R will similarly be unique for each copy of the circuit.

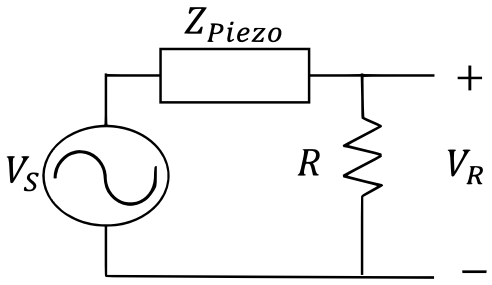

Fig. 3. Piezo Measurement Circuit

V_R was measured by using the 12-bit analog-to-digital converter (ADC) built into a EK-TM4C123GXL model Tiva LaunchPad. The use of a microcontroller allows for the majority of the response generation process to be automated. The downside to using this method is the ADC values can be noisy for singular readings. Additionally, an AC voltage cannot be directly digitized by the ADC. In order to obtain consistent measurements, the microcontroller samples the input voltage 10 times and determines what the peak reading was for those samples. This peak detection process is performed 10,000 times. Those 10,000 values are then averaged together to determine an overall average peak voltage value. By averaging so many values together, we are able to somewhat offset the noise and therefore increase the reliability. The number of samples per run (10) and number of runs (10,000) were experimentally determined for an input AC voltage of 300 KHz. The number of samples each run must make is heavily dependent on the frequency of the input voltage. As such, changes to only the sampling rate or the frequency of the input voltage are likely to have a negative impact on the reliability of the system.

C. Complete Architecture

The size of the 12-bit ADC potentially limits how many bits long each response from the PUF can be without introducing some form of padding such as feeding the 12-bit ADC value into a hash function. The proposed design accounts for this by taking measurements from eight instances of the circuit shown in Figure 3. Figure 4 shows a fully constructed PUF.

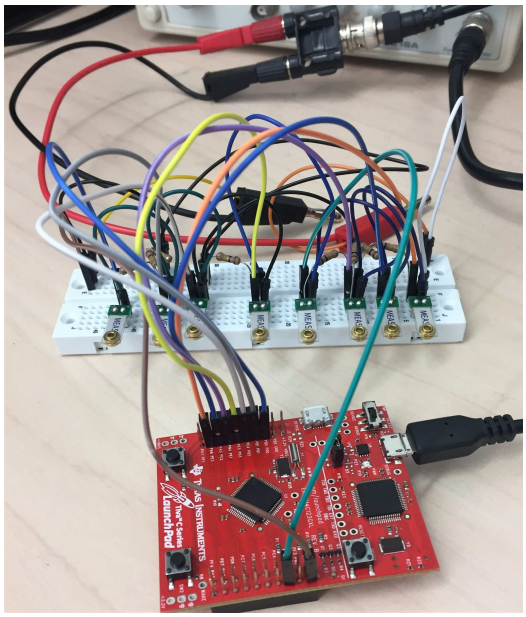

Fig. 4. Complete PUF Circuit

By default, the microcontroller has a base clock speed of 16 MHz and its ADC has a maximum sampling rate of 125K samples per second. For our implementation we chose to configure the microcontroller so that its clock speed was increased to 80 MHz and the maximum sampling rate of the ADC was increased to 1M samples per second. The main reason for the increases is the ADC is required to make a total of 100,000 samples per piezo sensor and 800,000 overall to generate a single response. Increasing the rates reduces the amount of time required to generate a response and also increases the accuracy when trying to calculate the peak voltage of the AC voltage input.

Rather than simply taking the readings from each circuit and combining them to generate a response, the proposed system compares the sum voltages for banks of three sensors and determines which one is larger. The result of the comparison is denoted by a single bit. The uniqueness of the piezo sensors due to process variations should result in unique voltage readings for each circuit. Creating summations of voltage readings for three different circuits greatly increases the number of possible unique comparison values. The added benefit to comparing readings is it should effectively be a comparison of the intrinsic characteristics that are unique to each circuit. Other factors such as the AC voltage source will cancel out assuming they uniformly affect each circuit. A total of 128 comparisons are made to generate the 128-bit response of the proposed PUF. Which sensors are compared and in what order is predetermined and will be fully explained in the next section.

D. Response Bit Calculation Algorithm

As previously described, the PUF calculates an average peak voltage associated with each piezo sensor. Three of those values are chosen, summed together, and then compared a summation of three different values. The result of that comparison is represented by a single response bit that is 1 if the first summation is larger, or 0 if it is not. This process is repeated 127 more times to generate a complete 128-bit response. Each comparison must be unique in terms of which groups of three are compared. This extends to preventing situations where the same groups are compared twice by reversing the comparison. For example, seeing if sensors zero, one, and two are greater than three, four, and five before later checking if sensors three, four, and five are greater than zero, one, and two. Each instance of this would effectively reduce the size of the response by one bit because two bits are complements of each other.

Additionally, special care must be given when choosing the combinations of circuits that are being summed and compared to avoid biasing the result. Consider, as an example, if the majority of the comparisons contained the third piezo sensor on the left hand side. If the value associated with that sensor happened to be the largest of all of the voltage values, then as a result the response bits should be biased towards 1. The values associated with each piezo sensor should be used 48 times on each side of the comparison. This was determined by multiplying the number of comparisons (128) by the number of values summed on each side (3) before dividing by the total number of sensors (8).

Rather than create a single algorithm for generating 128 bits from balanced comparisons, it was determined that a shorter algorithm could be used to generate a subset of 8 bits that was still balanced. Invoking the algorithm 16 times would then result in a 128-bit response that was not biased towards any single reading. Algorithm 1 shows the steps of this algorithm.

Algorithm 1 assumes that all 128 of the response bits are contained in an array $bits$. It will generate 8 response bits beginning at the location denoted by $place$. Array v contains the 8 peak voltage values associated with each piezo sensor (piezo sensor 0 is in location 0, sensor 1 is in location 1, etc.). Arrays l and r each denote the three piezo sensors whose associated values are to be summed together to make the first comparison. The generated response bit will be 1 if the sum of the values associated with l is greater than the sum of the values associated with r. Otherwise, the response bit will be 0. The value of $place$ is incremented by 1 after each bit is generated to keep track of which bit of the overall 128-bit response will be generated next. The determination of which sensors to use in each subsequent comparison occurs by incrementing each sensor by 1 and then rolling back to 0 if the result would have been 8. The process completes after 8 total comparisons have been made and as a result 8 response bits have been generated. This algorithm guarantees that the value associated with each piezo sensor will be used 3 times on each side of the comparison.

As previously mentioned, Algorithm 1 must be fed a series of 16 inputs in order to generate an entire 128-bit response. Each of these inputs must be chosen so that invoking Algorithm 1 does not inadvertently result in multiple instances of the same comparison. Algorithm 2 shows a list of input values $left$ and $right$ for the arrays $l[]$ and $r[]$, respectively, in Algorithm 1 that can be used to generate a 128-bit response without using the same comparison to generate multiple response bits.

Algorithm 1 PUF 8-bit Response Comparison Balancing Algorithm

```
1:  procedure BALANCE(bits, place, v[ ], l[ ], r[ ])
2:      bits ← Array containing response bits
3:      place ← Current response bit to be generated
4:      v[ ] ← Array of each circuit's peak voltage
5:      l[ ] ← Array of 3 circuits to be summed
6:      r[ ] ← Array of 3 circuits to be summed
7:      for i = 0; i < 8; i = i + 1 do {
8:          lsum = v[(i + l[0]) mod 8]
                 +v[(i + l[1]) mod 8]
                 +v[(i + l[2]) mod 8]
9:          rsum = v[(i + r[0]) mod 8]
                 +v[(i + r[1]) mod 8]
                 +v[(i + r[2]) mod 8]
10:         if lsum > rsum then
11:             bits[place] = 1
12:         else
13:             bits[place] = 0
14:         place=place+1
15:     }
16:     return
```

Algorithm 2 Input Values to Balancing Algorithm

1:	left: $0, 1, 2$	right: $3, 4, 5$
2:	left: $0, 1, 3$	right: $2, 4, 5$
3:	left: $0, 1, 4$	right: $2, 3, 5$
4:	left: $0, 1, 5$	right: $2, 3, 4$
5:	left: $0, 1, 6$	right: $2, 3, 4$
6:	left: $0, 1, 7$	right: $2, 3, 4$
7:	left: $0, 2, 3$	right: $1, 4, 5$
8:	left: $0, 2, 4$	right: $1, 3, 5$
9:	left: $0, 2, 5$	right: $1, 3, 4$
10:	left: $0, 2, 6$	right: $1, 3, 4$
11:	left: $0, 2, 7$	right: $1, 3, 4$
12:	left: $0, 3, 4$	right: $1, 2, 5$
13:	left: $0, 3, 5$	right: $1, 2, 6$
14:	left: $0, 3, 6$	right: $1, 2, 7$
15:	left: $0, 3, 7$	right: $1, 2, 4$
16:	left: $0, 4, 5$	right: $1, 2, 3$

The end result of invoking Algorithm 1 with the inputs shown in Algorithm 2 is a 128-bit response with improved uniformity due to the lack of bias towards any single piezo sensor. The values associated with each piezo sensor are used an equal number of times in both summations on either side of of the comparison on line 10 of Algorithm 1. During the generation of a 128-bit response, the peak voltage associated with each piezo sensor will be used a total of 96 times (48 times on each side of the comparison).

IV. Testing Configuration and Results

The physically unclonable function (PUF) proposed in this work was evaluated in terms of its reliability and uniformity as described in [24]. For testing, we created three copies of the proposed PUF. The average reliability and average uniformity was evaluated individually for each PUF and overall as a whole.

We did not evaluate our proposed PUF in terms of uniqueness, which serves as an indicator of how well an individual PUF can be distinguished from other copies of the PUF. We feel that our small sample size of three would prevent any uniqueness values from being truly meaningful. By comparison, uniqueness testing performed in existing literature can make use of simulated PUF copies to perform uniqueness testing for sample sizes that are orders of magnitude larger than ours.

A. Reliability Testing

The reliability of a PUF denotes how likely it is for a given input challenge to always produce the correct response. The ideal value for this property is 100% meaning that the PUF will always produce the correct response. The following equation, originally described in [24], can be used to calculate the reliability for a n-bit response:

$$Reliability = 100\% - \frac{1}{k}\sum_{i=1}^{k}\frac{HD(R_i, R_{i,t}^{'})}{n} \times 100\% \quad (1)$$

Where $HD(R_i, R_{i,t}^{'})$ represents the hamming distance (HD) between a reference response R_i and a response generated with different environmental conditions $R_{i,t}^{'}$. The variable k represents how many instances of the PUF there are.

We evaluated the reliability of our proposed PUF by recording the responses of each copy of the PUF over a period of 10 days. We used the Day 1 response as the reference response that all subsequent responses were compared to. Figure 5 shows a graph of the reliability values for each copy of the PUF. The worst case reliability for each PUF was observed to be 89.8%, 92.2%, and 98.4%. Table I contains the average reliability values for each copy of the PUF. The overall average reliability was determined to be 96.1%. Ideally, each copy of the PUF would have 100% reliability for each day. This was not the case for out tested PUFs. PUF3 was the PUF that came closest to having optimal reliability. The differences in results for each copy of the PUFs is purely random. They are a manifestation of the intrinsic variations within the piezo sensors that allow for them to be used to create the PUF in the first place. Each copy of the PUF is otherwise identical.

TABLE I. Average Reliability of Proposed PUF

PUF1	PUF2	PUF3	Total
91.84%	96.53%	99.83 %	96.07 %

Additionally, the reliability of a chosen PUF was tested across a range of temperatures from -20°C to 0°C and from 25°C to 80°C in increments of 5°C. 25°C was chosen as the reference point since it is room temperature. A freezer was used to generate temperatures from -20°C to 0°C and a temperature chamber was used for 25°C to 80°C. Due to limitations in the facilities available to us, we were not able to test over the range between 0°C and 25°C. Figure 6 shows the reliability graph with respect to temperature. The red line running between 0°C and 25°C is an interpolation between our measured values.

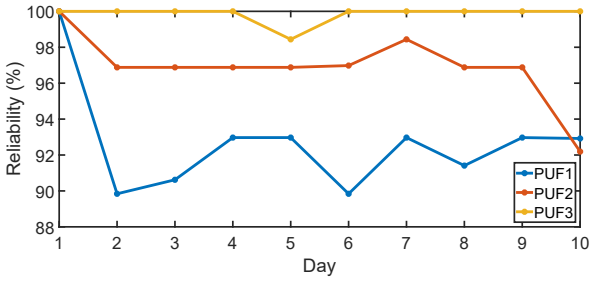

Fig. 5. Reliability Graph

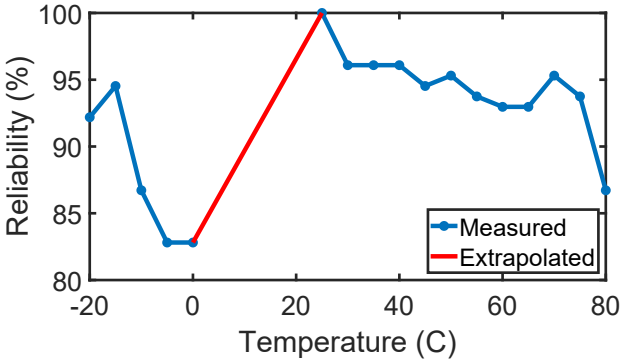

Fig. 6. Temperature Reliability Graph. The red line on the graph represents an extrapolation of the reliability values between 0°C and 25°C

The graph in Figure 6 shows that the reliability values decrease as the temperature move away from room temperature (25°C). The reliability still remains fairly consistent as the temperature rises above 25°C. Conversely, there is a drop at subzero temperatures that first appears to rise as the temperature continues to decrease from 0°C. The reliability then drops from -15°C to -20°C. This is likely just randomness of the PUF itself as its overall trend of the reliability decreasing as the temperature deviates from room temperature is typical of PUFs.

B. Uniformity Testing

The uniformity of a PUF is a measure of how balanced its generated responses are. An ideal PUF will have an equal number of 1's and 0's in the bits of its responses. Therefore, the ideal uniformity value is 50%. There following equation, originally described in [24], can be used to calculate uniformity:

$$Uniformity = \frac{1}{n}\sum_{l=1}^{n} R_{i,l} \times 100\% \quad (2)$$

Where $R_{i,l}$ is the l-th bit of a n-bit response generated by PUF i. Figure 7 shows the uniformity of each daily reading that was recorded for each copy of the PUF. Just like with the reliability testing, PUF3 was very consistent in its readings compared to the other copies. Even though there was more variation in the other copies, their readings were still reasonably close to the ideal uniformity value of 50%.

Uniformity can vary between each instance of a PUF due to intrinsic variations that are present despite each copy of the PUF being otherwise identical. For that reason we averaged the uniformity values of the responses from each PUF to obtain an idea of the general uniformity that can be expected from the proposed PUF. Table II shows the average uniformity for each of the proposed PUFs. The average uniformity for the individual PUFs were calculated by calculating and then averaging the uniformity of each response from the 10 day period previously used for the reliability calculations. The average uniformity of our PUF implementations was 47.52%.

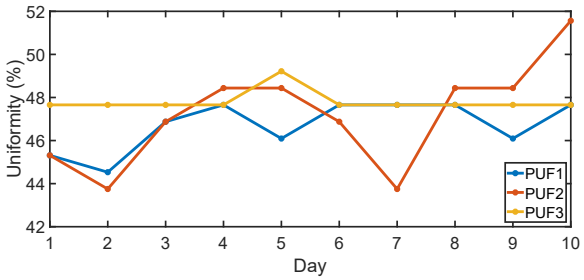

Fig. 7. Uniformity Graph

TABLE II. AVERAGE UNIFORMITY OF PROPOSED PUF

PUF1	PUF2	PUF3	Total
46.72%	47.19%	47.81 %	47.24 %

V. COMPARISON TO EXISTING DESIGNS

Using sensors as the basis for the design of our proposed PUF makes it somewhat difficult to compare to existing designs. During our research of existing literature, we found that works which described PUF designs that were not silicon based did not typically provide performance metrics like we provided in the previous section. For this reason, we are not able to make direct comparisons to other PUFs that are based on sensors. Instead, we can only highlight the functional advantages of our proposed design.

As previously mentioned in this paper, PUF designs have been proposed which are based on a range of devices including microelectromechanical systems (MEMS) based sensors [16] [17], device touchscreens [18], photodiodes [19], and solar cells [20]. The MEMs designs [16] [17] utilize a MEMs gyroscope to generate responses. This raises questions about how easily a given orientation of the gyroscope can be reproduced. A similar problem arises with the touchscreen based PUF [18]. It requires a user to trace a specified pattern with their finger. Even with it being a set pattern, error should still be introduced by a human trying to replicate fine motions used to trace the pattern. The issue with the proposed photodiode based PUF [19] is it actually requires a conventional PUF as part of its design. Lastly, the solar cell work [20] is not as fully formed as the other designs. It shows that solar cells have intrinsic variations that could be used as a PUF, but does not present a complete design. This information is summarized in Table III.

Our proposed design does not have any of these issues. Challenges can be easily reproduced for any copy of the PUF as the pattern of comparisons made are purely software based. In addition, our proposed design does not require an existing conventional PUF for proper operation. Currently, the downside to our proposed design is a sinusoidal input voltage is needed to really observe the unique properties of individual piezo sensors.

VI. CONCLUSIONS

Even though the results of testing were good, there are other areas for improvement that are worth exploring in future works as they could make our proposed design more suitable for inclusion in IoT applications. One such area of concern is the limited number of challenge-response pairs. By design, each copy of the proposed PUF should have only a single possible response. This is unfortunate because the PUF becomes effectively useless once that response is compromised. Increasing the number of challenge-response pairs would strengthen the PUF to where it would require a greater number of challenge-response pairs to be compromised before the PUF itself had to be replaced.

Another topic to explore is using the piezo sensors as a source of power in addition to a source of response bit generation. As currently configured, the microcontroller is not able to use the piezo sensors as source of power. A PUF that could act as even a partial source of power would be attractive for certain IoT applications as devices can be subjected to power constraints [1] [25].

In conclusion, we have proposed a method for using piezo sensors to create a physically unclonable function (PUF). The results of our initial rounds of testing are encouraging enough to indicate that our proposed method is viable way to create PUFs. The use of a microcontroller and energy harvesting devices further establishes the possibility of incorporating the proposed PUF into IoT devices as a cybersecurity solution.

ACKNOWLEDGEMENT

This research was partially supported by grant from National Science Foundation under Grant No: 1738662.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

TABLE III. PUF COMPARISON

PUF	Description	Drawback
MEMs [16] [17]	The response of a gyroscope is used to derive the PUF responses	Concerns over the reproducibility of the challenge.
Touchscreen	Mobile device app requires a user to trace a set pattern with their finger	Concerns over the reproducibility of the challenge.
Photodiode [19]	Summed voltages of two groups of solar cells are compared to generate response bits	Requires a conventional PUF to operate.
Solarcells [20]	Solar cells generate a unique response based on input light intensity and ambient temperature	Not yet a fully formed PUF.
Proposed Design	Uses a microcontroller to compare voltage readings across banks of piezo sensors	Requires an sinusoidal input source.

[2] Mcafee and the Center for Strategic and International Studies, "Economic impact of cybercrime - no slowing down," https://www.csis.org/analysis/economic-impact-cybercrime, accessed: 2018-4-12.

[3] ——, "Mcafee and CSIS: Stopping cybercrime can positively impact world economies," https://www.mcafee.com/us/about/news/2014/q2/20140609-01.aspx, accessed: 2018-4-12.

[4] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the internet of things: A standardization perspective," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 265–275, 2014.

[5] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[6] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[7] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer *et al.*, "Internet of things strategic research roadmap," *Internet of Things-Global Technological and Societal Trends*, vol. 1, no. 2011, pp. 9–52, 2011.

[8] M. Gorlatova, J. Sarik, G. Grebla, M. Cong, I. Kymissis, and G. Zussman, "Movers and shakers: Kinetic energy harvesting for the internet of things," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1. ACM, 2014, pp. 407–419.

[9] M. K. Stojčev, M. R. Kosanović, and L. R. Golubović, "Power management and energy harvesting techniques for wireless sensor nodes," in *Telecommunication in Modern Satellite, Cable, and Broadcasting Services, 2009. TELSIKS'09. 9th International Conference on*. IEEE, 2009, pp. 65–72.

[10] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004, pp. 176–179.

[11] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[12] B. L. P. Gassend, "Physical random functions," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[13] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *International workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2007, pp. 63–80.

[14] Y. Su, J. Holleman, and B. Otis, "A 1.6 pj/bit 96% stable chip-id generating circuit using process variations," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*. IEEE, 2007, pp. 406–611.

[15] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic pufs from flip-flops on reconfigurable devices," in *3rd Benelux workshop on information and system security (WISSec 2008)*, vol. 17, 2008, p. 2008.

[16] O. Willers, C. Huth, J. Guajardo, and H. Seidel, "Mems gyroscopes as physical unclonable functions," *IACR Cryptology ePrint Archive*, vol. 2016, p. 261, 2016.

[17] A. Aysu, N. F. Ghalaty, Z. Franklin, M. P. Yali, and P. Schaumont, "Digital fingerprints for low-cost platforms using mems sensors," in *Proceedings of the Workshop on Embedded Systems Security*. ACM, 2013, p. 2.

[18] R. A. Scheel and A. Tyagi, "Characterizing composite user-device touchscreen physical unclonable functions (pufs) for mobile device authentication," in *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*. ACM, 2015, pp. 3–13.

[19] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 112–117.

[20] E. Aponte, "A study on energy harvesters for physical unclonable functions and random number generation," Ph.D. dissertation, Virginia Tech, 2017.

[21] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[22] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual design automation conference*. ACM, 2007, pp. 9–14.

[23] M. Feiri, J. Petit, and F. Kargl, "Efficient and secure storage of private keys for pseudonymous vehicular communication," in *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*. ACM, 2013, pp. 9–18.

[24] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded systems design with FPGAs*. Springer, 2013, pp. 245–267.

[25] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

Carson Labrado received the B.S. degrees in Electrical Engineering and in Computer Engineering in 2014, and the M.S. degree in Electrical Engineering in 2017 from the University of Kentucky, Lexington, KY, USA. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Kentucky.

His Current research interests include physically unclonable functions and hardware security for IoT and vehicles.

Himanshu Thapliyal received the Ph.D. degree in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 2011.

He has authored over 100 journal/conference articles. He is currently an Assistant Professor and Endowed Robley D. Evans Faculty Fellow with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA. His current research interests include emerging nanotechnologies, low-power hardware security of IoT devices, and smart health.