2021

Energy Harvesting and Sensor Based Hardware Security Primitives for Cyber-Physical Systems

Carson Labrado
University of Kentucky, carson_labrado@hotmail.com
Digital Object Identifier: https://doi.org/10.13023/etd.2021.019

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Carson Labrado, Student

Dr. Himanshu Thapliyal, Major Professor

Dr. Daniel Lau, Director of Graduate Studies

ENERGY HARVESTING AND SENSOR BASED HARDWARE SECURITY PRIMITIVES FOR CYBER-PHYSICAL SYSTEMS

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in Electrical Engineering in the College of Engineering
at the University of Kentucky

By
Carson Joseph Labrado
Lexington, Kentucky

Director: Dr. Himanshu Thapliyal, Associate Professor of
Electrical and Computer Engineering
Lexington, Kentucky
2021

ABSTRACT OF DISSERTATION

ENERGY HARVESTING AND SENSOR BASED HARDWARE SECURITY PRIMITIVES FOR CYBER-PHYSICAL SYSTEMS

The last few decades have seen a large proliferation in the prevalence of cyber-physical systems. Although cyber-physical systems can offer numerous advantages to society, their large scale adoption does not come without risks. Internet of Things (IoT) devices can be considered a significant component within cyber-physical systems. They can provide network communication in addition to controlling the various sensors and actuators that exist within the larger cyber-physical system. The adoption of IoT features can also provide attackers with new potential avenues to access and exploit a system's vulnerabilities. Previously, existing systems could more or less be considered a closed system with few potential points of access for attackers. Security was thus not typically a core consideration when these systems were originally designed. The cumulative effect is that these systems are now vulnerable to new security risks without having native security countermeasures that can easily address these vulnerabilities. Even just adding standard security features to these systems is itself not a simple task. The devices that make up these systems tend to have strict resource constraints in the form of power consumption and processing power. In this dissertation, we explore how security devices known as Physically Unclonable Functions (PUFs) could be used to address these concerns.

PUFs are a class of circuits 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. Existing PUF designs are commonly based around transistor level variations in a special purpose integrated circuit (IC). Integrating these designs within a system would still require additional hardware along with system modification to interact with the device. We address these concerns by proposing a novel PUF design methodology for the creation of PUFs whose integration within these systems would minimize the cost of redesigning the system by reducing the need to add additional hardware. This goal is achieved by creating PUF designs from components that may already exist within these systems.

A PUF designed from existing components creates the possibility of adding a PUF (and thus security features) to the system without actually adding any additional hardware. This could allow PUFs to become a more attractive security option for integration with resource constrained devices. Our proposed approach specifically targets sensors and energy harvesting devices since they can provide core functions within cyber-physical systems such as power generation and sensing capabilities. These components are known to exhibit variations due to the manufacturing process and could thus be utilized to design a PUF. Our first contribution is the proposal of a novel PUF design methodology based on using components which are already commonly found within cyber-physical systems. The proposed methodology uses eight sensors or energy harvesting devices along with a microcontroller.

It is unlikely that single type of sensor or energy harvester will exist in all possible cyber-physical systems. Therefore, it is important to create a range of designs in order to reach a greater portion of cyber-physical systems. The second contribution of this work is the design of a PUF based on piezo sensors. Our third contribution is the design of a PUF that utilizes thermistor temperature sensors. The fourth contribution of this work is a proposed solar cell based PUF design. Furthermore, as a fifth contribution of this dissertation we evaluate a selection of common solar cell materials to establish which type of solar cell would be best suited to the creation of a PUF based on the operating conditions. The viability of the proposed designs is evaluated through testing in terms of reliability and uniformity. In addition, Monte Carlo simulations are performed to evaluate the uniqueness property of the designs.

For our final contribution we illustrate the security benefits that can be achieved through the adoption of PUFs by cyber-physical systems. For this purpose we chose to highlight vehicles since they are a very popular example of a cyber-physical system and they face unique security challenges which are not readily solvable by standard solutions. Our contribution is the proposal of a novel controller area network (CAN) security framework that is based on PUFs. The framework does not require any changes to the underlying CAN protocol and also minimizes the amount of additional message passing overhead needed for its operation. The proposed framework is a good example of how the cost associated with implementing such a framework could be further reduced through the adoption of our proposed PUF designs. The end result is a method which could introduce security to an inherently insecure system while also making its integration as seamless as possible by attempting to minimize the need for additional hardware.

KEYWORDS: Cyber-Physical Systems (CPS), Internet of Things (IoT), Cybersecurity, Hardware Security, Physically Unclonable Functions (PUFs), Vehicles

Carson Labrado

February 28^{th}, 2021
Date

ENERGY HARVESTING AND SENSOR BASED HARDWARE SECURITY PRIMITIVES FOR CYBER-PHYSICAL SYSTEMS

By

Carson Joseph Labrado

Dr. Himanshu Thapliyal
Director of Thesis

Dr. Daniel Lau
Director of Graduate Studies

February 28^{th}, 2021
Date

ACKNOWLEDGEMENTS

The research in this dissertation was partially supported by grants from Kentucky Science and Engineering Foundation per Grant Agreement KSEF-3998-RDE-020 and National Science Foundation under Grant No:1738662.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Cyber-Physical Systems

Researchers have made the argument that cyber-physical systems (CPS) and Internet of Things (IoT) have enough commonalities to be considered anywhere from partially overlapping fields [107] to effectively equivalent [55]. Other researchers have argued that IoT should be considered a subset of CPS [188] whereas others have argued it is the opposite such that CPS is actually a subset of IoT [134]. The lack of a clear separation points directly to just how interconnected the two terms actually are. This is further supported by the fact that a special publication from the National Institute of Standards and Technology (NIST) concluded that the definitions of cyber-physical systems and IoT are actually converging over time. Recognizing this unified perspective provides the opporunity for the various research communities to "work together to develop unified, new, hybrid, discrete and continuous methods for CPS and IoT design, operation, and assurance" [49].

For the purposes of this dissertation we consider IoT to be a subset of CPS as defined in [188]. That work defines CPS as the "merging and integration of Industry Control Systems, Critical Systems, Critical Infrastructures, Internet of Things (IoT) and Embedded Systems" [188]. Cyber-physical systems typically use a combination of sensors, actuators, processing units, and communication via a network. Some very common examples include vehicles and the aforementioned IoT devices. IoT devices are particularly noteworthy as they can exist as either a cyber-physical system, or as a component of a cyber-physical system. They can be considered a major component within certain cyber-physical systems as they can provide the network communication capabilities in addition to directly interfacing with the sensors and actuators. Due to being a key component, any IoT specific security issues should prove just as concerning for the cyber-physical system as a whole [57]. Figure 1.1 provides an example of the pivotal role that IoT devices can play within a larger cyber-physical system.

Cyber-physical systems cover an incredibly diverse range of devices in across a wide range of fields including medical devices, smart homes and cities, environmental monitoring, and industrial processes [27]. Attempting to provide security for these systems is a never ending process as new vulnerabilities are discovered and in return new solutions must be developed. The range of uses for cyber-physical systems can nearly eliminate the

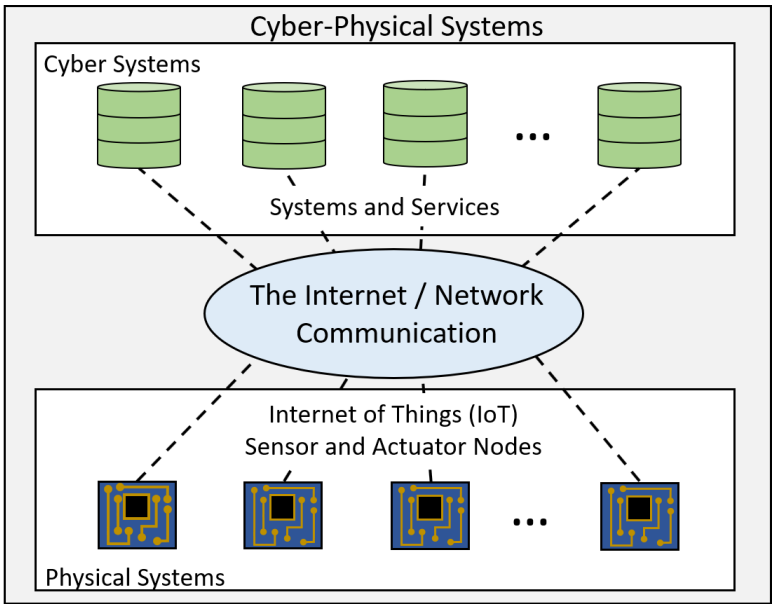


Figure 1.1: Example of CPS - IoT Integration

potential for one size fits all solutions. We believe one approach that can help address this issue is by focusing on providing security for very popular examples of cyber-physical systems such as vehicles and IoT devices. IoT devices are particularly noteworthy as they can be further used as components within other systems. While securing a main component within cyber-physical systems will not necessarily make the larger system completely secure, it will at least serve as a step in the right direction. It is for this reason that we take a particular interest in the security of IoT devices. In addition, IoT seemed to be the ideal component to highlight as it would appear to have the most widespread impact. This is due not only its already widespread adoption, but also its projected growth over the coming years.

1.2 Proliferation of IoT and Associated Security Concerns

Some researchers believe that the Internet of Things (IoT) will be the main component of the next era in computing [54]. 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.

Consider the communication capabilities of vehicles as a notable example. Vehicles have adopted increasingly complex features such as collision avoidance, infotainment options, traction control, remote start, vehicle-to-vehicle (V2V) communication, autonomous driving, etc. Vehicles contain numerous electronic control units (ECUs) to control these

various subsystems. All of the ECUs are connected by internal communication buses in order to exchange critical information between the subsystems. Additionally, some of these features require communication with external sources. A consequence of these features is the introduction of multiple potential access points that could be exploited by an attacker.

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. Previous reports from McAfee and the Center for Strategic and International Studies (CSIS) have shown a rising trend in the total global loss due to cyber crime [113] [114]. Attackers have shown that they are more than willing to make IoT devices the focus of their attacks. In the last few years for example, compromised IoT devices have been used to create botnets. Botnets are a network of compromised machines that can be used by an attacker for a variety of malicious purposes including distributed denial-of-service (DDoS) attacks, password cracking, and cryptocurrency mining. Once a machine is infected, it seeks to propagate the infection to other machines in its network by exploiting known vulnerabilities [19].

IoT devices would appear to present an ideal target due to a combination of their lack of security features and the sheer number of these devices that are currently in existence. The Mirai botnet in late 2016 was the first major botnet to be primarily composed of embedded and IoT devices. At its peak the botnet had infected 600 thousand devices [8]. One DDoS attack launched by the botnet was able to disrupt service to many prominent websites including Twitter, New York Times, Reddit, and Airbnb by targeting Domain Name Service (DNS) company Dyn [44]. A separate DDoS attack against French webhosting company OVH set the record for largest recorded DDoS attack with a size of at least 1.1 terabits per second (tbps) [168].

1.2.1 Common Security Threats and Challenges

The threats faced by IoT devices are just one, albeit very notable, example of the security threats that are facing cyber-physical systems as a whole. Other cyber-physical systems such as industrial control systems (ICS), smart grid, medical devices, and smart cars have been shown to be similarly vulnerable to attackers [63]. At a glance, there could appear to exist little commonality between these various types of cyber-physical systems. When it comes to security however, the devices in various fields are actually subject to a number of similar threats and constraints. Some examples of this include denial-of-service (DoS) attacks against ICS [142] and smart cars [20] and exploiting the lack of encryption in medical devices [81] and smart grids [16].

A recurring source of concern stems from the integration of IoT features. IoT devices are known to contain more security risks than conventional computing devices [19]. IoT devices can contain a multitude of vulnerabilities including insecure access interfaces, deployment locations that allow for easy unprotected physical access, and insufficient cryptographic mechanisms (including none at all in some cases) [18, 19, 79]. This is especially concerning when coupled with that fact that there are currently (as of 2018) 7 billion actively connected IoT devices (39.3% of all connected devices worldwide). These numbers are projected to grow to the point that in 2025 there will be 21.5 billion actively connected IoT devices worldwide. IoT devices would then represent 62.5% of all actively connected devices [106].

As originally designed, the dominant methods of security in these systems were related to "security by obscurity". Devices were assumed to operate in isolation, and it was therefore difficult for an attacker to access them. However, the push towards a connected world has resulted in many of these previously isolated devices now including support for external communication over a variety of networks. The increase in connectivity has also introduced several previously unconsidered possible attack vectors.

1.2.2 Possible Security Solutions

On the surface, just simply introducing more security features to these devices seems to be a reasonable approach to protect them from attackers. Unfortunately, these devices commonly have low power, small amounts of available memory, and limited processing capabilities. These factors can prove prohibitive to adding new security features. As a result, researchers have begun to place added emphasis on introducing security measures into IoT systems to help safeguard them from cyberattacks [78] [151] [191]. One area of research that has drawn attention as a potential cybercrime countermeasure for IoT devices is Physically Unclonable Functions (PUFs).

A PUF, which will be explained in-depth later in this dissertation, is a device that uses inherent variations caused by the manufacturing process to create unique and unclonable IDs. The PUF accepts an input "challenge" and in return outputs an associated "response". Due to intrinsic variations in the devices, a challenge that is given to two different copies of a PUF should result in different responses. This uniqueness property of PUFs can be seen in Figure 1.2. In the figure, the same challenge is sent to two copies of the same PUF. Each copy is a specific instance of a particular PUF design. The copies are therefore identical from a design standpoint. However, the outputs of the PUFs are not equivalent. This behavior can be utilized to implement various security measures.

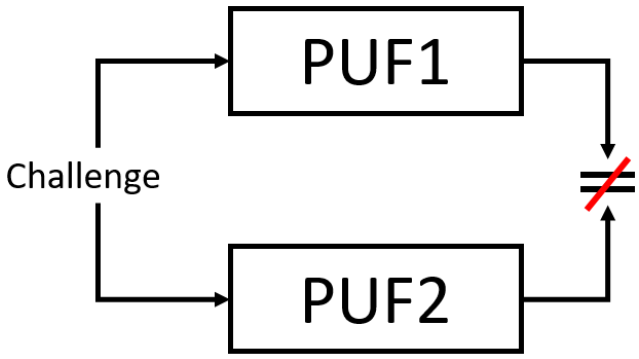


Figure 1.2: Uniqueness Property of PUF (previously published in [98])

Researchers have proposed PUF designs using a variety of techniques and implementation mediums. The common ground between the designs is leveraging intrinsic variations that should be unique to each instance of a device. Figure 1.3 contains a visual representation of the taxonomy of PUF designs along with examples of each.

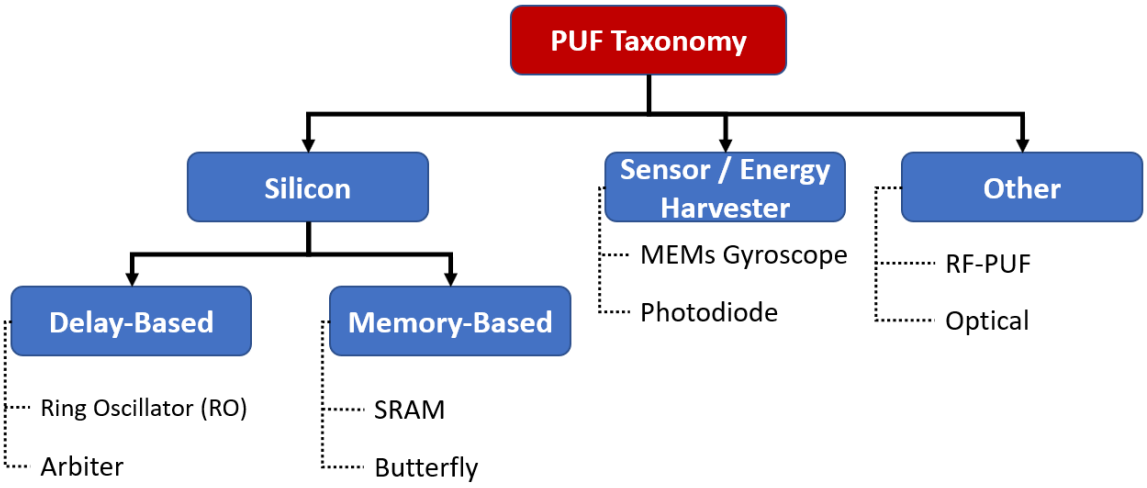


Figure 1.3: Taxonomy of PUF Designs

Silicon based PUFs are one popular medium. Silicon-based designs are based on transistor-level variations in gate delays or initial values of memory units. Ring Oscillator (RO) PUF [47] and Arbiter PUF [99] are notable delay-based PUFs while SRAM [53] and butterfly [53] are examples of memory-based designs. Other PUF designs are based around sensors or energy havesters. These components are designed to react to external conditions and in response generate an output which depending on the application can be used for sensing (sensors) or power generation (energy harvesters). A couple of examples include PUFs based on microelectromechanical systems (MEMS) gyroscopes [178] and photodiodes [139]. Other proposed PUF designs do not easily fit within the previously mentioned categories. One example is the RF-PUF which uses machine learning to identify the transmitter of a signal based on unique radio frequency (RF) properties [25]. Another example is the optical PUF which fires a laser at a transparent film [42]. The resulting speckle pattern will be unique to each film and can thus used to generate a response which in turn is unique to each copy of the PUF. All of these designs will be covered in more detail in the next chapter.

1.3 Motivation

Various cyber-physical systems including IoT devices and vehicles face some of the same challenges when it comes to implementing security measures. Namely, they tend to face similar resource constraints when compared to larger computing systems. This necessitates application specific implementation of security features. Due to the combination of real-time constraints and limited computational resources that are typical of these areas, researchers have sought ways to implement security primitives while minimizing the amount of additional hardware and computational resources that would be required.

Physically Unclonable Functions (PUFs) are one such area show promise as security solution. Previous works have shown how PUFs could be used to securely generate and

store secret keys [45, 155] while other works have proposed PUF-based security protocols for use in protecting sensor nodes [11] or securing radio-frequency identification (RFID) systems [17].

While PUFs could prove to be a novel security solution, their integration would not be completely seamless. PUFs are commonly based on CMOS ICs which require dedicated hardware to function correctly. This dedicated hardware results in additional costs in terms of hardware and power consumption. The performance of the device itself could suffer due to the additional overhead required to operate and/or communicate with the PUF. Therefore, a PUF designed for use in cyber-physical systems should also give special consideration towards reducing these costs as much as possible. The goal of this work is the creation of a class of PUFs whose integration with various sorts of cyber-physical devices would incur a minimal cost in the form of additional hardware.

One potential way to achieve this would be to try to limit the amount of special purpose hardware that is solely dedicated to implementing security features. This could help minimize both the costs of adding the hardware and the resources needed to interact with it. The core challenge then becomes how to remove hardware without similarly removing functionality. Our solution is to take the novel approach of designing PUFs from components that already exist within the device.

Designing a PUF from existing components would help address some of the concerns associated with integrating more established PUF designs. Additional physical hardware is potentially no longer necessary for the PUF's addition to the system. Furthermore, 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. The proposed approach should also have the added benefit of simplifying the PUF's future integration with these technologies since the required underlying hardware would in theory already be present.

When creating these novel PUF designs it is important to use components with as widespread adoption as possible. Otherwise the range of devices that could incorporate the proposed PUF without requiring any additional hardware could be severely limited. A given device can typically be expected to contain some combination processing units, sensors, and actuators. Furthermore those devices will occasionally be integrated with multiple types of energy harvesting devices including solar cells, thermoelectrics, and piezoelectric devices [172]. These devices were integrated so that they could be used in various applications such as power generation and sensing. The performance of these sensors and energy harvesting devices can vary between individual instances. This variance can be traced back to intrinsic variations in the components that are caused by the manufacturing process. These variations could potentially make the sensors and energy harvesters a good candidate for PUF creation.

By pursuing this approach, the ultimate end goal is creating the ability to add security features to devices and systems without needing to add any additional hardware. In effect, a device's existing energy harvesters and sensors serve as a source of security in addition to their original purpose of environmental monitoring and power generation. To broaden the appeal of using PUFs created from existing components, this dissertation proposes PUF designs which are derived from three different components: piezo sensors, thermistor temperature sensors, and solar cells. This approach allows the work to be applicable to a

much wider selection of cyber-physical devices than if we had based our proposed designs on only a single choice from the previously mentioned components. Figure 1.4 illustrates where our proposed PUF designs would be located within the broader PUF taxonomy that was previously shown in Figure 1.3.

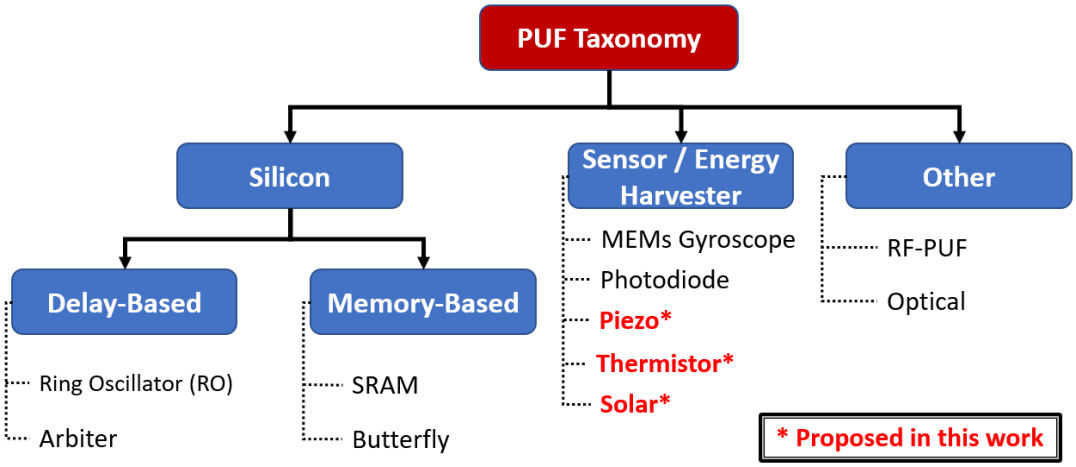


Figure 1.4: Classification of Proposed PUF Designs

As a proof of concept, we have created prototypes of the different proposed PUF designs and evaluated their performance metrics to measure their viability for use as a PUF. Each PUF design generates a single 128-bit response that should be unique to the specific PUF copy. The designs achieve this by directly leveraging the intrinsic variations between individual instances of each component. Furthermore, we demonstrate why PUFs should be seriously considered as an option to address known security vulnerabilities that do not already have easy and obvious solutions. This is expressed by considering the Controller Area Network (CAN) within vehicles as a case study. CAN was designed without any security considerations. Researchers have shown how its lack of built-in security features can be exploited to attack vehicles. We propose a security framework for the CAN bus that is based on PUFs. By leveraging the unique properties of PUFs our framework is able to add security features that by design are not previously supported. The use of PUFs allows us to avoid changing the CAN protocol itself and thus prevent the need for drastic redesigns of core vehicular infrastructure.

1.4 Contributions

The following contributions have been made so far in developing hardware security primitives for cyber-physical systems.

Contribution 1: Proposal of a novel physically unclonable function (PUF) response balancing algorithm The algorithm is able to securely generate a 128-bit response. It requires only eight copies of a given sensor or energy harvester. Each response bit represents the result of a comparison between groupings of three sensors or energy harvesters. A total of 128 comparisons are made in a predetermined pattern. Each comparison is unique so

that knowledge of bit location's value would not immediately reveal the value of any other bit locations.

Contribution 2: Proposal of novel piezo sensor based PUF circuit design methodology We have proposed a PUF design methodology that leverages the intrinsic variations piezo sensors to create a PUF which generates a 128-bit response. The sensors can be represented by an equivalent circuit whose impedance will differ due to intrinsic variations. We measure the impedance values by applying an AC voltage. We created three prototypes and evaluated their performance over a temperature range of -20°C to 0°C and 25°C to 80°C.

Contribution 3: Proposal of novel thermistor temperature sensor based PUF circuit design methodology We have proposed a PUF design methodology that leverages the intrinsic variations in thermistor temperature sensors to create a PUF which generates a 128-bit response. The sensors directly react to the ambient temperature by varying their resistance. Intrinsic variations will produce differences in resistance despite having the same external stimuli. We measure those differences via a voltage dividing circuit. We created five prototypes and evaluated their performance over both a temperature range of -20°C to 80°C and relative humidity of 30% to 100%. We also performed Monte Carlo Simulations on 1000 simulated PUF instances to evaluate the uniqueness.

Contribution 4: Proposal of novel solar cell based PUF circuit design methodology We have proposed a PUF design methodology that leverages the intrinsic variations in solar cells to create a PUF. Each solar cell generates a voltage based on the intensity of the light shining on it. These voltages will differ between cells due to intrinsic variations. The proposed PUF generates a 128-bit response. The proposed design should show resiliency to changes in temperature and light intensity while also being flexible enough to work with a variety of solar cells.

Contribution 5: Evaluation of solar cell materials for the creation of novel PUF circuits We evaluated three common solar cell materials to determine their suitability for the creation of PUFs. The solar cell materials materials were monocrystalline sillicone, polycrystalline silicon, and amorphous silicon. We created three copies of each of the three different solar cell based designs for a total of nine PUF copies to be evaluated. Performance was evaluated over a temperature range of -20°C to 80°C and relative humidity of 30% to 100% and a light intensity range of 40 Watts/m^2 to 90 Watts/m^2. We were also able to perform Monte Carlo Simulations on 1000 simulated instances to evaluate the uniqueness of one of the materials.

Contribution 6: Proposal of a novel PUF-Based Controller Area Network (CAN) security framework We demonstrate how PUFs could be integrated into a a cyber-physical system such as a vehicular CAN bus to provide security features for communication between nodes. Our proposed framework does not require any modification to the existing CAN protocol. Our novel approach also significantly reduces overhead in the form of both the amount of CAN frames and the amount of time required for operation when compared to existing approaches.

1.5 Dissertation Outline

This dissertation is organized as follows: Chapter 2 provides background information on security approaches in cyber-physical systems as well as specifics about PUF-based approaches; Chapter 3 explains our proposed method for creating a PUF from piezo sensors; Chapter 4 presents our thermistor temperature sensor based PUF design methodology; Chapter 5 presents our solar cell based PUF designs while also providing a relative performance evaluation of different solar cell materials; Chapter 6 explains how leveraging PUFs could help improve vehicular security and presents our PUF-based security framework for a vehicular CAN bus; Chapter 7 concludes this work and provides some possible directions for future research.

The work presented in Chapter 3 was previously published in [94] "C. Labrado and H. Thapliyal, Design of a Piezoelectric-Based Physically Unclonable Function for IoT Security, IEEE Internet of Things Journal © 2018 IEEE". The work published in Chapter 4 was previously published in [98]. The work in Chapter 5 was previously published in [93] "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd". The work in Chapter 6 has been accepted for publication in ACM JETC [97]. All of these works have contributed to the first two chapters in varying capacities. In additon, some portions of Chapters 1 and 2 were previously published in [96] "C. Labrado and H. Thapliyal, Hardware Security Primitives for Vehicles, IEEE Consumer Electronics Magazine © 2019 IEEE". A provisional patent has been granted for the design methodology [161]. The piezo sensor work was also a hardware demonstration at a conference [95] and the security framework has been accepted for a hardware demonstration in the upcoming iteration of the conference.

Chapter 2

Background and Related Work

In this chapter we will discuss various aspects of security as they relate to IoT and vehicles and highlight some of the proposed hardware solutions. The information presented is intended to provide a general understanding of current hardware security approaches in the field. Some popular areas of vehicular security research such as modifying network protocols [51] [169] [52] [102] [157] and the creation of Intrusion Detection Systems (IDS) [122] [30] [125] will mostly not be covered as they do not necessarily fall under the category of hardware solutions. We will solely focus on the the topics that are more closely hardware aspects of security. This information includes security properties, cryptographic algorithms, and hardware solutions. Later sections in this chapter will focus on various aspects of PUFs including various design methodology, evaluation metrics, and security applications.

2.1 Security Properties

Security itself can be described as consisting of five basic properties: confidentiality, integrity, authenticity, availability, and non-repudiation. A system that is completely secure will be able to guarantee all five of the properties. These properties are described below in terms of messages being sent between two parties within a communication system:

- Confidentiality - This is the guarantee that no outside party is able to determine the contents of messages sent during communication between two corresponding parties. The implementation of this property prevents any third party from eavesdropping on a conversation. A common method for implementing this property is to use a chosen cryptosystem to encrypt messages between two parties. In their encrypted state, the messages should not reveal any information about the actual content of the message. Only the communicating parties should be able to decrypt the encrypted messages. Therefore, as long as attackers cannot decrypt the encrypted messages they are effectively prevented from eavesdropping.

- Integrity - This is the guarantee that all received messages are correct. The original contents of the messages have not been alterted or tampered with in any way. If any tampering has occurred, then it should be detected by the system. As will

be explained below, this property tends to be closely associated with authenticity. Hash functions, which will also be explained further down, are one somewhat simple method for implementing integrity.

- Authenticity - This is the guarantee that all communicating entities are who they claim to be. The correct application of this property will protect against a malicious third party attempting to forge messages or impersonate one of the communicating parties. This property is closely associated with integrity because it many cases it is somewhat useless to implement one property without the other. For example, there is not much utility in knowing a message came from a specific person (authenticity) without also knowing that the message has not been tampered with (integrity).

- Availability - This is the guarantee that the communication system will always be available and able to deliver messages between two communicating parties. A common type of attack against this property is a denial of service (DoS) attack in which an attacker attempts to restrict or even completely halt the availability of the overall communication system. This is accomplished by attempting to overload the system through flooding it with illegitimate messages to the point that legitimate messages are either delivered too slow to be useful or simply not delivered at all. Attacks that send illegitimate message from multiple sources are known as distributed denial of service (DDoS) flooding attacks [189].

- Non-Repudiation - This is the guarantee that either party is not able to deny any aspect of its communication. This means that if one party sent or received a message then it cannot deny that the message was sent or received, respectively. In some respects this property is more focused on protecting the communicating parties from each other, rather than protecting both parties from a malicious third party. One example of this property is it would be impossible for one party to send a malicious message and then deny it was the one who sent the message. Possible implementations of this property can include the creation of message signatures that are generated by a trusted third party [87].

Guaranteeing these properties can be very complex and costly. These difficulties can be further exacerbated when these properties are trying to be introduced to technologies and protocols that were not originally designed with security in mind. The CAN bus for example does not guarantee any of these properties [60] and can therefore present a major security risk that is not easily fixed.

2.2 Cryptographic Algorithms

Any description of security is likely to include a mention of cryptography. Cryptographic algorithms form the basis of cryptosystems and can allow data to be stored and transmitted in a secure format. A strong cryptographic algorithm will be based on a mathematical problem which is considered to be computationally infeasible. A well known example is the RSA cryptosystem [137] which is based around the difficulty in factoring large numbers.

The most common types of cryptographic algorithms for encryption and decryption are symmetric-key and asymmetric-key. In addition to those two types of algorithms, there are also secure hash functions which can have their own unique applications. A visual representation of these categories is shown in Figure 2.1.

The computational resources (power, memory, speed, etc.) available in IoT and vehicles can be somewhat less when compared to what is available in a conventional computer system. These reduced resources can place constraints on what cryptographic algorithms can be used in these resource constrained applications. For this reason, a relatively new field of cryptosystems called lightweight cryptography (LWC) has been created [148] [115]. LWC schemes are designed in ways that are designed to reduce the amount of memory, latency, power, etc. required for computation. This is typically realized by reducing key sizes, simplifying the number of rounds in block ciphers, and reducing the complexity of required operations.

All of the algorithms that will be mentioned in this section are examples of cryptosystems that have been implemented in various existing security schemes.

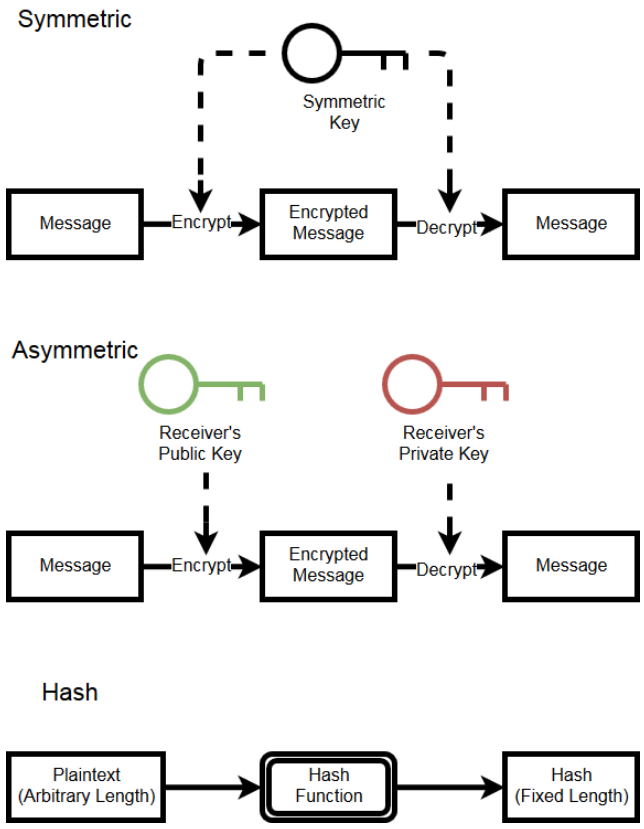


Figure 2.1: Categories of Cryptographic Algorithms

2.2.1 Symmetric-key Cryptography

Symmetric-key cryptosystems are so named because they use the same key for the encryption and decryption of messages. This is a straightforward approach, but it does have the issue where the sender and the receiver of each message must already have obtained a copy of key through some type of secure channel. The Advanced Encryption Standard (AES) algorithm is a commonly used symmetric-key cryptosystem for transmitting information over insecure channels. AES allows for key sizes of 128, 192, and 256 bits. Another downside to symmetric-key cryptography is those methods can require generating a new key for every message.

2.2.2 Asymmetric-key Cryptography

Asymmetric-key cryptosystems (also known as public key) use different keys for the encryption and decryption of messages. The encryption key is publicly displayed so that any entity may use it. The decryption key is kept private and is therefore only known by its owner. There is a mathematical relationship between the keys that allows the public key to be easily generated from the private key, but not vice versa. This eliminates the need in symmetric-key cryptosystems for either using secure channels to share keys or generating a new shared key for each message. The aforementioned RSA cryptosystem is a public key cryptosystem based on factoring large numbers. Elliptic-curve cryptography (ECC) is another approach which is based on elliptic curves. The Diffie-Hellman key exchange [117] protocol has a variation based on elliptic curves called Elliptic-curve Diffie-Hellman (ECDH).

2.2.3 Hash Functions

Hash functions take an input of arbitrary length and outputs a result of fixed length. Because the output is a fixed length, there is the possibility of a collision occurring in which two different inputs result in the same output hash. A strong hash function will have very small odds of a collision occurring and it will be designed in such a way that the smallest alteration to the input will produce a wildly different output hash. Hash functions are one-way functions so they can not be used in a similar method to symmetric-key and asymmetric-key cryptography. Hash functions can instead be used for verification purposes. The simplest application of hash functions is verifying the integrity of data. Hash functions can take inputs of arbitrary size so a large set of data could be hashed to generate a much smaller hash output. The smallest change to the data should result in a wildly different hash output. In this way, data can be checked for unauthorized tampering by having to only compare hash outputs rather than the entirety of the original data input. WHIRLPOOL [15] is one popular hash function used today. WHIRLPOOL takes an input message less than 2^{256} bits long and produces an output that is 512 bits. The Secure Hash Algorithm (SHA) family of hash function are another example of commonly used hash functions.

2.3 Vehicular Communication Networks

Modern vehicles continue to become smarter and more complex with increasing arrays of sensors and electrical devices that must be able to communicate with each other. The burgeoning emergence of a larger ecosystem of autonomous vehicles and smart cities will give rise to an additional need for vehicles to have external communication abilities.

2.3.1 Internal Communication

Multiple types of communication networks have been proposed and or implemented to allow the various vehicular hardware devices such as electronic control units (ECUs) and sensors to communicate in a fast and efficient manner. We will discuss some of the more popular network standards such as FlexRay, (Media Oriented Systems Transport (MOST), Local Interconnect Network (LIN), and the previously mentioned Controller Area Network (CAN) bus. More in-depth descriptions of these networks and other existing networks can be found in existing works [128] [164].

- CAN (Controller Area Network) - CAN was developed in the 1980's as a bus system to simplify the process of connecting multiple ECUs within a vehicle. Previously, vehicles used dedicated point-to-point wiring which begins to result in large and expensive wiring harnesses as more and more electronic devices are added to the vehicle. The introduction of CAN allowed manufacturers to reduce the wiring cost, complexity, and weight of vehicular systems [65]. The general utility and cost effective performance of CAN has allowed it to remain widely used to this day.

- FlexRay - As vehicular technology continued to evolve, the performance requirements of new advanced control and safety systems were beginning to exceed CAN's capabilities. FlexRay was designed to meet the performance requirements of both these new systems and future systems that will emerge in the next generation of vehicles [66]. Current applications of FlexRay are performance critical subsystems such as adaptive cruise control, high-performance powertrain, and anti-lock braking systems (ABS) [66] [111]. The increased performance of FlexRay, however is offset by FlexRay's increased cost when compared to CAN. At least for now, FlexRay is only a partial successor to CAN as the complete replacement of CAN with FlexRay in vehicular networks would be cost prohibitive. CAN will remain the preferred protocol in subsystems whose performance requirements do not exceed CAN's capabilities.

- MOST (Media Oriented Systems Transport) - MOST was developed by the MOST Cooperation in 1998 as a communication network to be primarily used in communication between multimedia devices found in infotainment systems within vehicles. These multimedia devices such as radios, GPS navigation, video displays, and entertainment systems tend require large amounts of data which facilitated the need for a multimedia-centric network [126]. Previously, the only versions of the MOST protocol were MOST25 and MOST50. The newer version, MOST150, has since

been introduced and has superior performance capabilities (e.g. speed and bandwidth) when compared to the existing version. In addition to improved performance, MOST150 added support for Ethernet packets and MAC addressing [85] [34].

- LIN (Local Interconnect Network) - LIN is a somewhat newer type of bus protocol which was first developed via a collaboration between automakers Audi, BMW, DaimlerChrysler, Volkswagen, and Volvo [140] as a low-cost network for use in non-critical subsystems. These non-critical subsystems do not require the same measure of speed and reliability as do the more performance-intensive subsystems that are typically connected via the CAN bus. By using LIN for minor subsystems in conjunction with CAN, manufacturors are able to save costs while in effect not truly impacting the functionality of these subsystems. Applications of LIN tend to include somewhat simple mechatronic subsystems like door locks, trunk releases, and seat controls [126].

2.3.2 External Communication

Advances in vehicular technology are trending toward the need for the creation of vehicular ad hoc networks (VANETs). VANETs are an emerging type of communication network that allow for the integration of vehicles and other entities such as Road-Side Infrastructure (RSI) into an overall communication network. These networks have a myriad of uses such as communication between autonomous vehicles and smart traffic through communication between transit authorities and vehicles. In the future, these networks could be used to aid autonomous vehicles in navigating roadways while requiring little to no input from the vehicles occupants. Much like with the internal communication networks, these VANETs have their own security vulnerabilities which much be addressed to provide protection against attacks. Further information on VANETs including their implementations, security vulnerabilities, and potential security solutions can be found in the following works [165] [43] [136] [6].

2.4 Hardware Security Modules

2.4.1 Trusted Platform Modules

Trusted Platform Modules (TPM) are cryptographic co-processors designed to integrate security into larger computer systems. They are now commonly included in computers as a secure hardware solution to fulfill security needs. The specifications for a TPM were standardized by ISO/IEC 11889 [150]. TPMs are self-contained in their operation meaning they are not reliant on the operating system used by the overall computer system to operate. This prevents the TPM from being compromised by any security vulnerabilities that might be present in the computer's operating system or applications. The security provided by a TPM is based around keys and as such the TPM provides multiple crypto engines to allow for symmetric and asymmetric key generation, encryption, and decryption. TPMs include other hardware features such as a random number generator and secure non-volatile storage. The current version of TPM is 2.0, which required the inclusion of new cryptographic

algorithms. Previous TPM versions only required RSA and the SHA-1 hash function. TPM 2.0 requires those algorithms in addition to AES-128 for symmetric cryptography, ECC algorithms for asymmetric cryptography, and the SHA-256 hashing function. Figure 2.2 contains a representation of the basic components that will be generally included in a given TPM 2.0 implementation. Several of the security features provided by TPMs for computer systems are also desirable in vehicular communication systems. Unfortunately, the integration of TPMs and vehicles can be more complicated than a direct inclusion of existing TPMs into vehicles. The unique challenges of vehicular security necessitate the development of TPMs that are specially designed for use in vehicles.

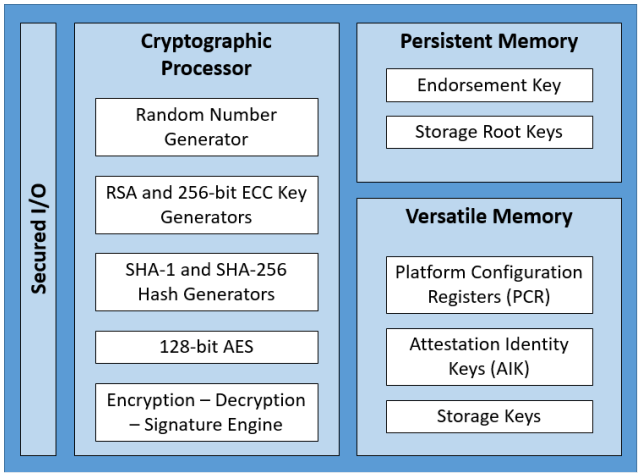


Figure 2.2: TPM 2.0 Components (© 2019 IEEE)

2.4.2 Vehicular Hardware Modules

One notable area of vehicular security research is the creation of security modules that can be introduced to the unsecured buses found in vehicles. These modules attempt to implement a measure of security without incurring significant costs. The additional security hardware can be designed for operation either within a chosen communication network, or as a gateway of sorts between multiple networks. These modules aim to provide a secure hardware to facilitate secure communication between nodes in both internal and external vehicular communication networks. In addition, the modules are capable of generating and storing cryptographic keys. The unique security challenges facing vehicles prevent the direct adaptation of existing trusted platform modules (TPMs) from normal computers to vehicles. Hardware security modules for vehicles instead must be specially designed so that the modules may be effectively integrated into vehicular communication networks. However, this does not mean that vehicular TPMs will be completely unique from existing implementations. Both existing TPMs and their proposed vehicular implementations tend to share multiple hardware components such as storage mediums and crypto engines.

Modules Within Singular Networks

The Controller Area Network (CAN) bus allows multiple ECUs to communicate with each other. CAN was not originally designed to include any sort of security features. This presents a major security concern as the ECUs connected to the CAN bus tend to be associated with critical systems such as braking and steering. An attacker who was able to gain access to the CAN bus would have numerous options for causing harm to a vehicle, its occupants, and other motorists. Modifying the protocols used by the communication networks has drawn interest from researchers as a method for introducing security [51] [169] [52] [102] [157]. The downside to modifying the protocol is it introduces the potential to increase the amount of computations that must be performed by the system. This increase could reduce the speed of communication between components and overall harm the performance of the system and as a result may not be adopted by manufacturers.

Other approaches have focused on the inclusion of new hardware. Researchers in [179] have proposed a vehicular hardware security module. A diagram of the module's architecture is shown in Figure 2.3. Their module is a general-purpose cryptographic co-processor designed for use in vehicles. The module can be combined with the individual ECUs to provide protection by connecting the proposed module to an in-vehicle communication network. The proposed hardware security module (HSM) was designed to include dedicated resources for the full range of cryptographic algorithms (asymmetric cryptographic, symmetric cryptography, and hash functions). The asymmetric crypto engine is ECC-256 and is commonly used for the creation and verification of digital signatures. The symmetric crypto engine used is AES-128 and allows the HSM to provide symmetric encryption and decryption. WHIRLPOOL is the cryptographic hash function provided by the HSM. The inclusion of a hash function allows the HSM to both generate and verify "fingerprints" such as plain hash and hash-based message authentication code (HMACs). The HSM also includes a pseudo-random number generator (PRNG) that provides pseudo-random numbers to be used during normal operations. The PRNG generates these numbers from an internal algorithm that can be seeded either internally by a physical true random number generator (TRNG) or by an external source such as an external TRNG during production of the HSM itself. Internally, the HSM has both unsecured RAM to act as a key buffer and nonvolitile memory (NVM) to allow for secure storage of keys. These features are kept behind a so-called cryptographic boundary that is separate from the actual application core that communicates with other ECUs attached to the in-vehicle communication system. The decision to include multiple crypto engines effectively allows the module to act as a general-purpose crypto-processor which can be reused in a variety of applications. The module shown in Figure 2.3 is referred to as "full" because it has everything needed to secure vehicle-to-everything (V2X) communications. Additionally, the researchers in [179] proposed variations of the module known as "medium" and "light". The "medium" variation is focused only on the security of communication within the vehicle. The "light" variation is designed to only secure interactions between ECUs and the sensors and actuators.

A different type of hardware module is one that is an Intrusion Detection System (IDS) [32]. When inserted into the CAN network, this module is able to identify all of the ECUs based on the inimitable characteristics of their signals. The module would then be able to

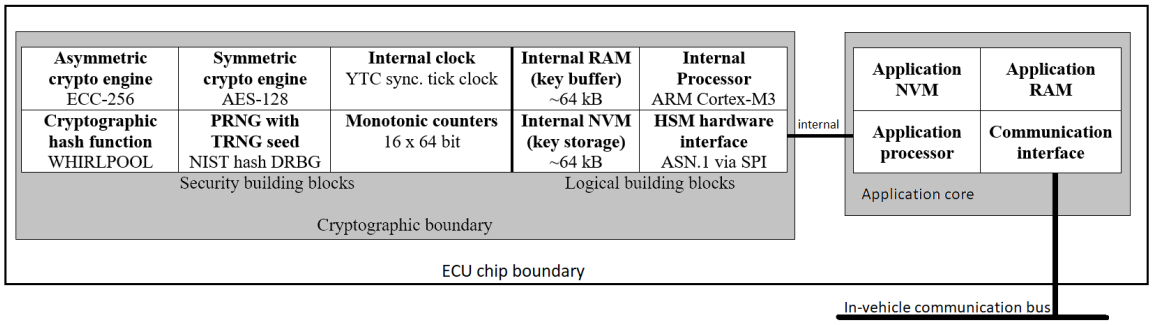


Figure 2.3: Hardware Security Module (HSM) Architecture from [179]

identify malicious ECUs that have been inserted into the network. What sets this IDS apart from other implementations is it would not require any change to the existing hardware of the ECUs and it does not actually require modifications to the CAN protocol. The main drawback to this module is the protocol used by the ECUs must still be changed. ECUs in the network would have to use the extended frame format of CAN. ECUs would require a firmware update to switch from the normal frame format that they currently use in order to be compatible with the proposed IDS.

Modules Between Networks

A slightly higher level form of security module is the introduction of a security gateway that allows for secure communication between different networks within the vehicle. An example of this method can be found in [180]. In their proposed system, each valid controller within a communication network is assigned a security certificate which has been digitally signed by the original equipment manufacturer (OEM). The gateway is able to use the public key which is associated with the OEM to verify the validity of the controllers. Messages sent from invalid controllers then have the option of either being processed separately, or just simply discarded.

2.5 Security Attacks and Countermeasures

This section will only address a selection of the types security attacks that are relevant to the hardware primitives that were discussed in previous sections. This is intended to be more of a general overview of the types of security vulnerabilities present in each field. More thorough and specific explanations can be found in existing works which are solely dedicated to this topic such as [123], [184] and [177] for IoT, and [153], [3], and [83] for vehicles.

Providing security is a challenging problem that is not readily solvable by conventional solutions. Even just the diagnosis of security threats has required the development of novel intrusion detection methods [121]. However, the security issues in sub fields such as those facing vehicles vehicles are not as isolated as they might appear. Similar security concerns are actually being raised in a variety of other areas. This general trend is a direct response

to society's adoption of the Internet of Things (IoT). The addition of smart features to an increasing number of consumer electronics has also introduced security vulnerabilities and concerns that were not present during the designing of the original devices.

Developing methods to combat these new challenges has drawn interest from a number of researchers. This has included classical approaches such as designing hardware security chips for mobile devices [74] and secure firmware validation and update schemes for personal home devices [31]. Other researchers have even explored novel PUF-based solutions such as creating device authentication schemes for IoT-enabled medical devices [187] and radio-frequency (RF) communication between nodes in a wireless network [24]. A method has even been proposed for integrating PUFs with blockchain technology to make it suitable for providing security in IoT devices [120].

Despite interest from researchers, the level of security preparations are still not sufficient to match the security vulnerabilities that will arise with both the continuing conversion of previously offline consumer electronics int IoT-enabled devices and advances in the sophistication of a device's IoT-compatible functionality. The task is a daunting one and we believe novel technologies such as PUFs have the potential to make a much larger impact in this area than they have made in security solutions in more classical systems. A later chapter in this work provides an example by showing how the inclusion of PUFs can introduce security features into a intra-vehicle network while minimizing any changes in its normal operation.

2.5.1 IoT Security Concerns

The design and use of IoT devices introduce security vulnerabilities that are not present in classical computing systems. One source of these vulnerabilities is IoT devices typically have limited computing resources. A second source of vulnerabilities is those devices can almost be commonly found in relatively insecure locations. Their placement in these locations can make it much easier for an attacker to physically access the device [184].

Limited Computing Resources

IoT devices tend to have limited computing resources. This serves as a bit of a hurdle when it comes to data protection. Various types of cryptographic schemes are usually used for data protection. Unfortunately, the limited resources available in IoT devices can make it difficult to implement popular encryption schemes while still meeting the real-time requirements that characterize many IoT applications. For this reason, researchers have proposed lightweight cryptography (LWC) which is desired specifically for implementation in resource constrained devices [148] [115]. LWC schemes attempt to provide an acceptable level of cryptographic security while reducing the amount of computational resources that are required to implement it in comparison to cryptosystems that are commonly used in applications that have more computational resources available. This is typically achieved by making design decision that include some combination of reducing key sizes, reducing the complexity of required intermediate operations, and reducing the number of rounds in block ciphers. PRESENT [21] and CLEFIA [146] are two examples of LWC families of block ciphers.

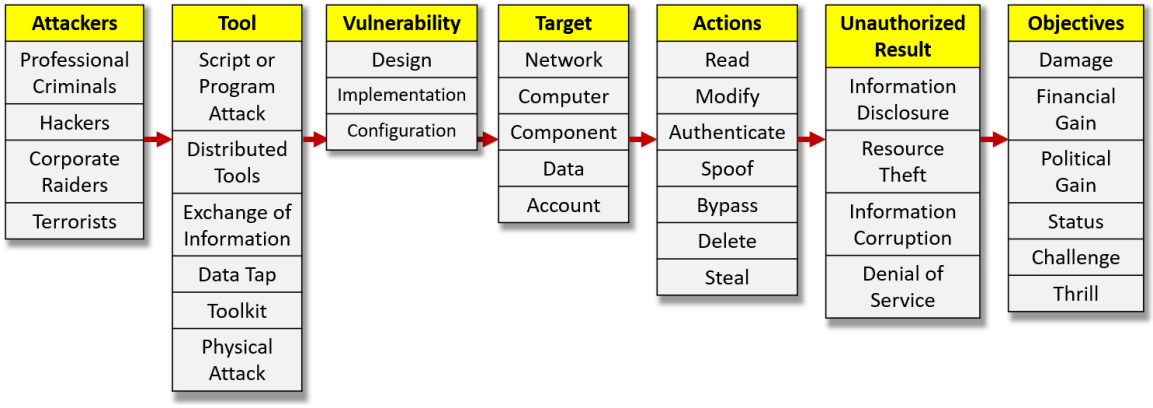


Figure 2.4: Taxonomy of Vehicular Security Attacks as Described in [22]

Insecure Location

IoT devices can often be placed in locations that are easily accessible by attackers. This means an attacker gaining physical access to a device should be treated as a realistic threat. With physical access to a device, an attacker could attempt to reverse engineer cryptographic and/or processing units on the device. As a countermeasure, researchers have developed hardware obfuscation techniques to mask the operations being carried out by those units and therefore make them resistant to reverse engineering. One method that has been proposed involves dynamically changing the order in which a processing unit receives data [41]. This makes what operations are currently being performed less obvious to the attacker. Other researchers have proposed using the responses from PUFs to either swap internal wirings [176] or reconfigure internal logic [185].

2.5.2 Vehicular Security Concerns

Figure 2.4 shows an attack taxonomy from [22] that has been modified for vehicles. The taxonomy provides the general outline of an attack including who the attackers could be, what tools they might use in the attack, the actions taken with those tools, and the attackers' overall objective for the attack. For the sake of brevity in this section, "communication between parties" is used in the general sense to refer to any type of vehicular communication that could occur such as between ECUs connected via a bus or between vehicles via a wireless network.

Eavesdropping Attacks

These types of attacks involve an attacker "listening in" to communications between two parties. In the context of vehicular security this could involve listening to communication channels such as the CAN bus or intra-vehicle messages. If these channels are insecure then an attacker could obtain sensitive information being passed through the network or could determine how to forge communications to impersonate one of the parties. The common countermeasure to eavesdropping has been to make sure all communications are encrypted

so attackers and other third parties are unable to gain any useful information. Implementing message encryption in these networks can be difficult due to limitations that are intrinsic to their design, such as messages being broadcast to all connected nodes. The integration of Trusted Platform Modules (TPMs) is one possible solution. These modules would provide all of the hardware resources necessary to integrate multiple types of cryptosystems. Physically unclonable functions (PUFs) could be useful as a way of both securely storing and generating keys to be used by the cryptosystem.

Data Tampering Attacks

These attacks occur when an attacker is able to modify messages between two parties without being detected. Consequences of this attack could include altering sensor readings within a vehicle or changing relative location information sent by a vehicle. Just like with eavesdropping, message encryption can help prevent tampering attacks.

Impersonation/Forgery Attacks

These attacks are when an attacker is able to successfully impersonate a communicating party. These attacks are potentially more severe than data tampering attacks because the attacker is able to completely forge new messages rather than only being able to alter existing messages. For example, an attacker could request sensitive information that would compromise the security of the vehicle. The attacked vehicle believes it is communicating with a trusted party and unknowingly sends the requested information straight to the attacker. Some counter measures include employing public-key cryptosystems and trusted-third parties like certificate authorities that can verify the authenticity of a message's author.

Man in the Middle Attacks

In some respects this attack is a combination of the previous attacks. In this attack scenario an attacker is able to both intercept and forward messages between two parties. The two parties are not aware that their messages are being routed through a third party. The parties become vulnerable to all of the previously mentioned attacks. The attacker also has the option of rerouting messages or selectively preventing certain messages from reaching their intended recipient. These attacks take on more significance in vehicular ad hoc networks (VANETs) due to intra-vehicle communication, the variety of information that might need to be shared, and the consequences that could result. A strong countermeasure to these attacks would be using a protocol that is non-forwardable thereby preventing the basis for this attack.

Denial of Service (DoS) Attacks

The purpose of this attack is to prevent the normal use of a network or service. This is commonly achieved by flooding the target with enough illegitimate messages to cause legitimate messages to be slowly delivered if not dropped altogether. In-vehicle bus communication networks like CAN are particularly vulnerable due to their broadcast nature and lack of a method of establishing authenticity. A countermeasure against DoS attacks

would be introducing a method for verifying the authenticity of nodes in the network [156]. Unverified nodes could be suspended from the network.

2.6 Physically Unclonable Functions

First proposed in [131], 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 2.5.

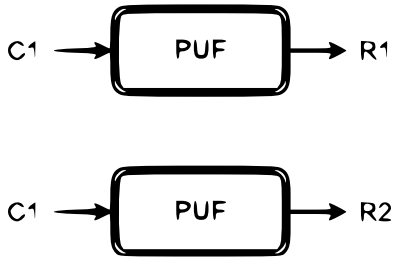


Figure 2.5: Example of Uniqueness Property of PUF (© 2018 IEEE)

In Figure 2.5 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 the Figure, 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. A PUF should be well designed enough that there is a minimal chance of CPR overlap between instances of a given PUF. For example, if a weak PUF produces a single 128-bit response, then the odds of having two PUFs both produce the exact same response should be 1 in 2^{128}. This is similar for strong PUFs except with the added possibility that copies of a given PUF could have some common CRPs. These concerns are related to a PUF's uniqueness property which will be discussed later in this section.

The inherent properties of PUFs make them attractive for use in security applications. 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 explored incorporating PUFs into a wide range of areas including key storage and generation, signature creation, and authenticity verification.

PUFs have commonly been implemented on application specific integrated circuits (ASICs) [190] or on field programmable gate arrays (FPGAs) [53] as transistor level device variations caused by the IC manufacturing process have been observed to be suitable for creating PUFs. Additional implementations have been proposed which are based on a wide range of devices including microelectromechanical systems (MEMS) based sensors [178] [12], device touchscreens [143], photodiodes [139], and solar cells [10].

2.6.1 PUF Evaluation Metrics

The following metrics are important measures of a PUF's performance. As the designs proposed in this dissertation are all weak PUFs, the equations and their explanations are provided in the context of evaluating weak PUF designs which produce a single response.

Reliability Testing

The reliability of a PUF is a measure of how well it will produce the correct response. The ideal reliability value of a PUF is 100%. This indicates that the PUF will never produce an erroneous response to a given challenge. Anything less than 100% is an indication that some portion of bits in the response will be flipped from their correct value. For exampe, if a PUF has a reliability of 95% then 5% of the bits in the response will be erroenous while the other 95% will be correct. The following equation (first described in [110]) is used to calculate the reliability for a n-bit response of:

$$Reliability = 100\% - \frac{1}{m}\sum_{t=1}^{m}\frac{HD(R_i, R_{i,t}^{'})}{n} \times 100\% \tag{2.1}$$

In this equation, R_i is a chosen reference response produced by a specific PUF instance i. $R_i^{'}$ is a response generated under different environmental conditions. A total of m responses are collected with different environmental conditions. Each of these generated responses ($R_{i,1}^{'}$, , $R_{i,2}^{'}$, ... ,, $R_{i,m}^{'}$) are compared to the reference response R_i. $HD(R_i, R_{i,t}^{'})$ is the hamming distance (HD) between the reference response (R_i) and the t-th generated response ($R_{i,t}^{'}$) of PUF i.

Uniformity Testing

Another standard PUF metric is uniformity. The uniformity of a PUF describes how "balanced" its responses are, i.e., what is the prevalence of 1's vs. 0's in the bits of the responses. Ideally, there will be an equal number of 1's and 0's to maximize the entrophy and thus maximimize the difficulty for an attacker trying to guess the value of a given bit. This ideal scenario is represented by a uniformity value of 50%. The following equation (first described in [110]) is used to calculate the uniformity of a n-bit response:

$$Uniformity = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} \times 100\% \tag{2.2}$$

In the above equation, $r_{i,l}$ represents the l-th bit of a n-bit long response generated by PUF instance i. In order to obtain a general uniformity of PUF we averaged together all the readings for a given test.

Uniqueness Testing

As described in [110], the uniqueness of a PUF represents the ability to distinguish one particular instance of a PUF from a group of PUFs of the same type. The ideal uniqueness value is 50%. The following equation is used to calculate uniqueness:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \tag{2.3}$$

The above equation determines the average hamming distance (HD) among k total PUFs. R_i and R_j represent n-bit responses produced by PUFs i and j, respectively where $i \neq j$.

It should be noted evaluating the uniqueness of a PUF design typically requires orders of magnitude more copies of a PUF to be created than what are feasible in a research context. For this reason, it is common practice to perform Monte Carlo simulations to generate enough unique simulated PUF copies to have a sufficiently large enough population to perform uniqueness testing. The simulated PUF copies are directly created from data describing the physical characteristics of the PUFs. That data should allow for the creation of a normal population of simulated devices that would accurately match what the actual physical population should be.

2.6.2 PUF Design Taxonomy

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 [99] [101], ring oscillator [47], SRAM [53], butterfly [53], latch [154], and flip-flop [108].

- 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 delay 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.

Non-Silicon Based PUFs

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 [178] [12], device touchscreens [143], photodiodes [139], and solar cells [10]. The MEMs designs [178] [12] utilize a MEMs gyroscope to generate responses. The touchscreen based PUF [143] 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 [139] 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 [10] shows that solar cells have intrinsic variations that could be used as a PUF, but does not actually present a complete design. One proposed authentication scheme [42] uses an optical PUF. This type of PUF involves firing a laser at a transparent film and recording the scattered speckles. Process variations will cause microscopic differences in the films and result in unique speckle patterns for each film and therefore a unique response for each copy of the PUF. Other researchers have proposed a PUF based on radio frequency (RF) communication [25]. The design uses a deep neural network at the receiver (Rx) to identify the unique properties of the signal's transmitter (Tx). The unique properties are a direct manifestation of process variations inherent to each transmitter.

2.6.3 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 to 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.

2.6.4 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. This is of course assuming the attacker would not be able to somehow get a copy of a PUF's response(s) from another device in the system. Creating a clone of the PUF in that way is only made possible by a vulnerability in the system that is completely unrelated to the PUF itself. 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 [155] [45]. 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. The reliability issues can be addressed by employing methods such as error correcting codes (ECCs) that are designed to improve reliability [33] [175]. Furthermore, PUF responses can 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. The origin of the challenge would be implementation dependent. Some possible sources could be other nodes seeking to communicate, some sort of central authority, or the challenge even could be hard coded into the device. 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.

2.7 Security Applications of PUFs

Physical Unclonable Functions (PUFs) have many potential uses in IoT and vehicular security. The unique attributes that are intrinsic to PUFs allow them to be used in ways that would be otherwise impossible to implement using classical technology. This section will describe just some of the existing approaches for applying to IoT security or vehicular security. It should be noted that even though the approaches in this section were explicitly designed for use either in IoT devices or vehicles, there exists enough of a similarity between the two fields that only minimal effort would be required for some of the proposed methods such as key storage to become applicable for both IoT applications and vehicular applications.

2.7.1 IoT Applications

Use of PUF in Hardware Obfuscation

As previously mentioned, some researchers have explored using PUFs as a method for obscuring the hardware of IoT devices to protect them from reverse engineering attacks. One such approach is to use the response from the PUF to swap internal wiring within the processing unit that is being protected [176]. A similar approach is the use the output of the PUF to actually obscure the internal logic instead of the actual wiring [185]. In this design, the original circuitry to be obscured is replaced with a PUF and some programmable fabric. The input data to the original circuit servers as challenge to the PUF and the programmable fabric is used as a PUF to allow the new circuit to function as originally intended.

PUF in FPGAs for IoT

Field programmable gate arrays (FPGAs) with Dynamic Partial Reconfiguration (DPR) capabilities can easibly be partially reconfigured. This makes them an attractive option for IoT applications as these devices have the added flexibility of having a minimally invasive method for updating the hardware. Unforunately, FPGAs that are DPR enabled can be subjected to hardware trojan insertions [73]. Some researchers have proposed using PUFs as way to fight against these attacks [72]. The researchers propose schemes that use the challenge response pairs of PUFs in individual FPGAs to control the DPR functionality. In the scheme a given FPGA creates a signature from challenges and responses and sends it to an authority to validate its identity and request permission to enable or disable DPR. By preventing and identifying unauthorized DPR usage, the device is able to protect itself against hardware trojans that seek to exploit this vulnerability.

2.7.2 Vehicular Applications

PUF as a Secure Storage Method

Researchers in [45] have proposed using PUFs as a method for storing private keys for use in vehicular communication. The typical methods of securely storing keys typically involve

the use of secure memory. Their method allows for secure key storage in completely unsecured memory. This is achieved by having the keys be derived from the responses of a PUF. The use of PUF ensures that an attacker would have to obtain the actual physical device in order to extract the key. Another work proposed storage methods that involve using either a strong PUF or a weak PUF [45]. For a strong PUF, multiple challenges and other helper data required to ultimately derive the key is stored in normal unsecured memory. Whenever a key is needed, the associated challenge is accessed and applied to the PUF. The desired key is then derived from the PUF response. This storage method is shown in Figure 2.6. For a weak PUF, only a single challenge and associated helper data is stored in unsecured memory. The PUF response to that challenge is used to derive a "master seed". That master seed can then be used to further derive multiple keys. This type of storage method is shown in Figure 2.7. In both scenarios, the security of the keys does not come from using secured memory, but rather the fact that the keys can not be derived without access to the actual PUF.

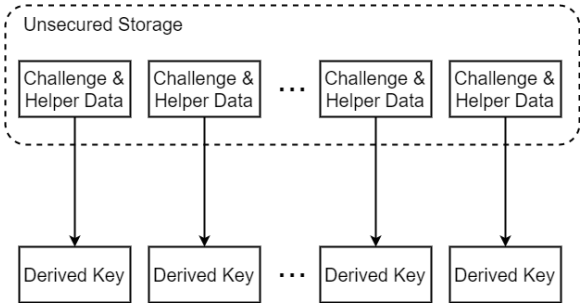


Figure 2.6: Strong PUF Key Storage Method from [45]

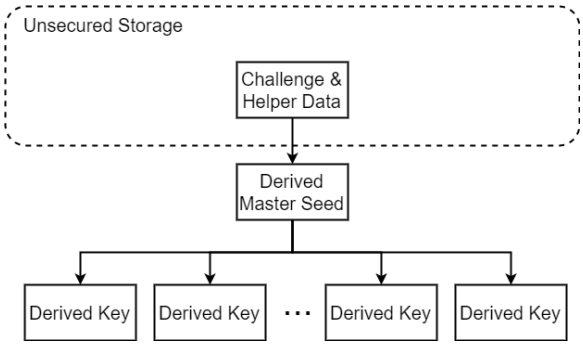


Figure 2.7: Weak PUF Key Storage Method from [45]

An alternative storage method has been proposed in [133]. In this method the PUF response itself is used to derive a pseudonym in the form of a public key and private key pair. An overall Certificate Authority (CA) issues a certificate as proof that it has verified the validity of the generated pseudonym. This certificate can then be included in communications with an external entity.

Use of PUFs in Communication

Approaches have been proposed in the literature for integrating PUFs with vehicular communication systems. One such approach lays out a method in which an optical PUF could be used as part of a non-forwardable authentication scheme for vehicle-to-vehicle communication [42]. Another approach is to incorporate a PUF into individual ECUs [156]. The use of PUF allows for the creation of an authentication method for ECUs that are attached to the CAN bus. A built-in authentication method makes it much harder for an attacker to insert a malicious ECU into one of the communication networks on the vehicle. This would help to eliminate the threat of an attacker attaching a malicious ECU to the network that could then be used to send erroneous messages within the network.

PUF in Vehicle to Vehicle Communication

Researchers have proposed a way to use PUFs as part of a larger communication system that is capable of avoiding adversary coalition attacks [42]. In a coalition attack, two or more adversaries impersonate the sender and receiver by intercepting messages and then forwarding them to their intended recipients. In the presented scenario [42], a given vehicle is communicating with other nearby vehicles. The recipients of these communications are authenticated using sensors to optically bind each communication with the correct vehicle. Visually binding to other vehicles allows the vehicle to know the relative location of each vehicle with which it is communicating. This knowledge of the exact location of nearby vehicles can be leveraged to aid in the vehicle responding to warning messages from said vehicles. By forwarding messages, adversaries can trick targeted vehicles into visually binding a communication to the adversary intercepting and forwarding messages instead of the actual intended recipient of the messages. Even if the adversaries in a coalition attack simply forward the intercepted messages without any sort of tampering, the attacked vehicle can still misidentify an adversary as the vehicle with which it is communicating. This could prove dangerous when attacked vehicles attempt to respond to warnings about emergency maneuvers such as emergency braking. The solution put forth by the researchers for mitigating these sorts of attacks is to create a message authentication method that is non-forwardable and therefore not susceptible to coalition attacks. Vehicles are assigned certificates from a trusted Certificate Authority (CA) which contain physical characteristic to aid in identifying the vehicle and challenge response pairs (CRPs). A vehicle wishing to establish communication sends its certificate to a given vehicle. The receiving vehicle extracts the information from the certificate. The receiving vehicle configures a laser to the challenge parameters contained in the certificate. The receiving vehicle uses its laser to stimulate the optical PUF on the sending vehicle and records the response. The receiving vehicle is able to authenticate the sending vehicle if the recorded response sufficiently matches the response contained in the certificate. The receiving vehicle then sends its certificate to the sending vehicle and the process is repeated with swapped roles to allow the sending vehicle to authenticate the receiving vehicle.

Integration of PUFs and ECUs

Researchers in [156] present a way of preventing Denial of Service (DoS) attacks within the CAN bus by implementing the ability to authenticate connected ECUs. The proposed method is based on assigning an ID to each ECU connected to the CAN bus. Each of the IDs are associated with specific challenge-response pairs (CRPs) of the PUFs that are incorporated into each ECU. A centralized reference monitor (RM) securely stores copies of the IDs and CRPs for each ECU within a trusted platform module (TPM). All of this data is determined and loaded when the vehicle itself is being built. Any ECU wishing to communicate along the CAN bus must first be authenticated by the RM. The authentication process, in summary, is essentially any ECU that wants to communicate with another ECU first sends the message to the RM along with its ID and a nonce, which is an arbitrarily chosen number that will only be used one time. The RM replies to the ECU with the challenge associated with the given ID and temporarily stores the nonce for a later comparison. The ECU receives the challenge from RM and computes its corresponding response. The ECU then replies with the challenge response, ID, and a copy of the previously sent nonce. The RM first compares the newly received nonce with the previously received nonce. The RM waits a set amount of time for a response for a response containing a matching nonce. If no response containing a message response is received during that time frame then the RM drops the original message. Once the RM receives a reply with a matching nonce it then determines if the challenge response is either a match or close enough to the stored challenge response associated with the ECU's ID. An appropriate challenge response allows the RM to determine the authenticity of the ECU and forward the ECU's original message to its intended destination. Challenge responses that are determined by the RM to not be valid result in the RM deciding the ECU is illegitimate and suspending it.

Chapter 3

Design of Piezo Sensor Based Physically Unclonable Function

3.1 Introduction

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. For our purposes those existing components should be ones that are already common in cyber-physical systems 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 [172]. For the purposes of this chapter 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 [48] [152]. In this chapter we propose a PUF design that is specifically targeted for use in cyber-physical systems. Our proposed design is constructed from components that are common in IoT devices such as microcontrollers and piezo sensors.

This chapter is organized as follows: Section II describes the methodology that was used to design our proposed PUF; Section III explains the testing metrics of reliability and uniformity and the results of those tests; Section IV compares our proposed design to existing sensor based PUF designs; and lastly, Section V concludes the chapter. Material from this chapter was previously published in [94] "C. Labrado and H. Thapliyal, Design of a Piezoelectric-Based Physically Unclonable Function for IoT Security, IEEE Internet of Things Journal © 2018 IEEE"

3.2 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. The response generation algorithm which will be described later in this section is responsible for ensuring that it is only the intrinsic characteristics that are being compared.

3.2.1 Piezo Sensor

As described in [10], a piezo sensor can be modeled by the Butterworth-van-Dyke equivalent circuit shown in Figure 3.1. 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.

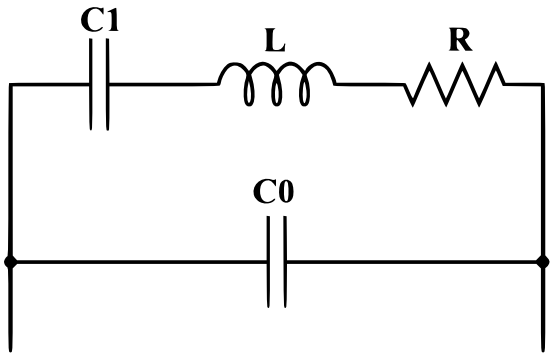


Figure 3.1: Piezo Sensor Butterworth-van-Dyke Equivalent Circuit (© 2018 IEEE)

3.2.2 Basic Piezo Circuit Diagram

The circuit shown in Figure 3.2 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. The piezo sensors used were the cantiliver-type MiniSense 100 Vibration Sensor from Measurement Specialties [149]. 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. 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.

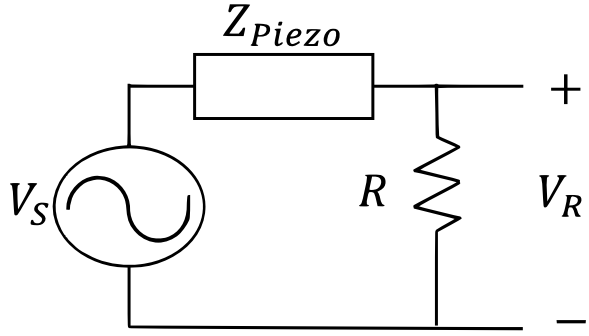


Figure 3.2: Piezo Measurement Circuit (© 2018 IEEE)

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.

3.2.3 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 3.2. Figure 3.3 shows a fully constructed PUF.

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. The default ADC sampling speed is slower than the 300 KHz AC voltage driving our circuit. For our implementation we chose to configure the microcontroller so that its clock speed was increased to 80 MHz and the

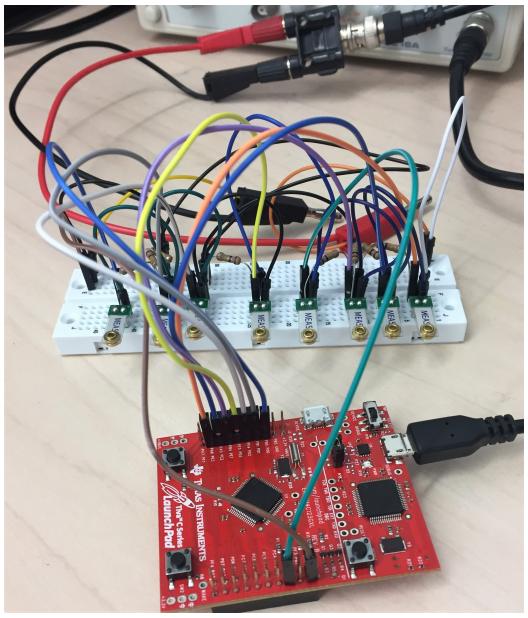


Figure 3.3: Complete PUF Circuit (© 2018 IEEE)

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.

3.2.4 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 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 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 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).

3.3 Testing Configuration and Results

The physically unclonable function (PUF) proposed in this chapter was evaluated in terms of its reliability and uniformity as described in [110]. 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.

3.3.1 Reliability Testing

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 3.4 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 3.1 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.

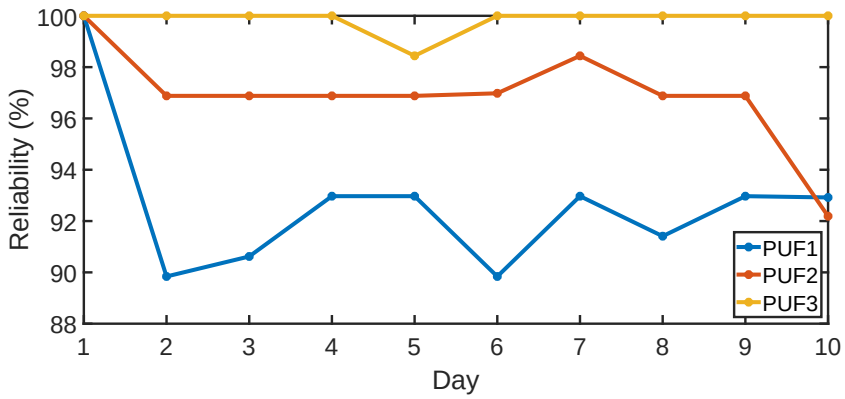


Figure 3.4: Reliability Graph (© 2018 IEEE)

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

Table 3.1: Average Reliability of Proposed PUF (© 2018 IEEE)

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

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 3.5 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.

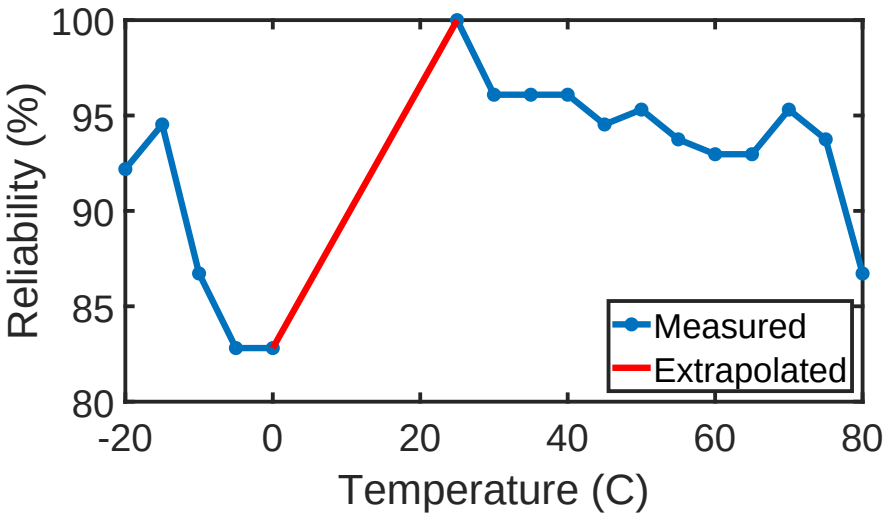


Figure 3.5: Temperature Reliability Graph. The red line on the graph represents an extrapolation of the reliability values between 0°C and 25°C (© 2018 IEEE)

The graph in Figure 3.5 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.

3.3.2 Uniformity Testing

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 3.2 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%.

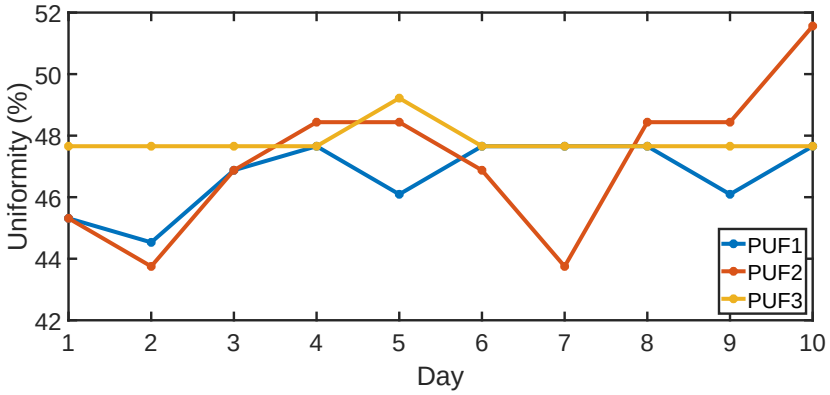


Figure 3.6: Uniformity Graph (© 2018 IEEE)

Table 3.2: Average Uniformity of Proposed PUF (© 2018 IEEE)

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

3.4 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 chapter, PUF designs have been proposed which are based on a range of devices including microelectromechanical systems (MEMS) based sensors [178] [12], device touchscreens [143], photodiodes [139], and solar cells [10]. The MEMs designs [178] [12] 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 [143]. 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 [139] is it actually requires a conventional PUF as part of its design. Lastly, the solar cell work [10] 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 3.3.

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.

Table 3.3: PUF Comparison (© 2018 IEEE)

PUF	Description	Drawback
MEMs [178] [12]	The response of a gyroscope is used to derive the PUF responses	Concerns over the reproducibility of the challenge.
Touchscreen [143]	Mobile device app requires a user to trace a set pattern with their finger	Concerns over the reproducibility of the challenge.
Photodiode [139]	Summed voltages of two groups of photodiodes are compared to generate response bits	Requires a conventional PUF to operate.
Solarcells [10]	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 a sinusoidal input source.

3.5 Conclusions

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 or vehicles as a cybersecurity solution.

Chapter 4

Use of Thermistor Temperature Sensors for Cyber-Physical System Security

4.1 Introduction

The previous chapter demonstrated how piezo sensors have been used to create a weak PUF design. However, the use of piezo sensors required including an AC voltage source in the design which further harms its utility. One potential solution to this issue is to explore creating a PUF from another component that is commonly found in cyber-physical systems. An example is thermistor temperature sensors. Thermistors have widespread appeal as shown by the presence of temperature sensing capabilities in a wide range of fields including health care [88], agriculture [116], and smart home environments [105]. In this chapter we propose a methodology that allows for using thermistor temperature sensors to create a PUF that is specifically targeted for application in cyber-physical systems. Our proposed design uses a microcontroller and thermistors which are themselves commonly used by these types of devices.

The rest of this chapter is organized as follows: Section 4.2 covers PUFs including security applications and design approaches that are relevant to our proposed design; Section 4.3 describes the design methodology behind our proposed PUF; Section 4.4 describes the tests used to evaluate our proposed PUF and presents the results of those tests; Section 4.5 compares our proposed PUF to existing sensor-based PUF designs; and finally, Section 4.6 concludes the chapter by providing a summary of our results. We evaluated the viability of using thermistors as a basis for creating a PUF by testing copies of the proposed design in terms of reliability and uniformity and used Monte Carlo simulations to evaluate the uniqueness. We provide the following:

- Proposal of a PUF circuit design methodology based on intrinsic variations between thermistors.
- Testing the proposed PUF's reliability over 1000 consecutive readings.
- Testing the proposed PUF's uniformity over 1000 consecutive readings.
- Testing the proposed PUF's reliability over a temperature range of -20 °C to 80 °C.

- Testing the proposed PUF's reliability over a relative humidity range of 30% to 100%
- Calculating the proposed PUF's uniqueness through Monte Carlo simulations on 1000 simulated instances.

The information in this chapter was previously published in [98].

4.2 Background and Related Work

This section provides information on PUFs including different design approaches and examples of their usage in security applications.

4.2.1 Physically Unclonable Functions

PUFs are a type of device that are commonly used in security applications. PUFs take a given "challenge" or input and use it to produce an associated "response" or output. A challenge and its associated response are collectively referred to as a challenge-response pair (CRP). PUFs are especially designed in a way that make them impossible to clone, hence the name "physically uncloneable". PUF operations rely on their own intrinsic variations that are commonly introduced during the manufacturing process. These variations are random and result in each instance of a given PUF with unique CRPs.

Additionally, PUFs can be characterized as either "weak" or "strong". Weak PUFs are characterized as having a very limited number of challenge-response pairs (CRPs), typically just one. They are used in applications where attackers are assumed to not be able to access the responses as knowing just one CRP could be enough to compromise it. Conversely, strong PUFs have a very large number of CRPs. This allows them to be used in applications where an attacker could obtain access to some of the CRPs. This is because strong PUFs should have enough possible challenges that an attacker will not be able to determine all possible CRPs if given a subset of CRPs.

4.2.2 Use of PUF as a Security Measure

The intrinsic properties of PUFs make them well suited to a variety of security applications. Each instance of a PUF should be both unique and unclonable. This places an extra hurdle in the way of attackers that forces them to obtain the actual PUF that is being targeted in the attack as it should be impossible for them to create an exact copy of the PUF. Researchers have begun proposing a wide range of security measures that seek to directly leverage the unique features of PUFs.

One major focus of research has been using PUFs as a way of securely generating and storing secret keys [45, 155]. The response from a PUF is used as a seed to generate secret keys. For weak PUFs, the response is a master seed from which all generated secret keys are ultimately derived. The downside to this approach is an attacker only must compromise a single CRP or potentially even one of the keys to compromise all the keys generated by the PUF. Using a strong PUF instead provides more security as each key is derived from a

different CRP. The CRPs of strong PUFs are unpredictable and therefore even if an attacker compromises some of the CRPs or keys it has generated, the rest are virtually unaffected.

An approach similar to the one used in key generation can be applied to remove the need for secure memory to store secret keys [45]. Compared to normal unsecure memory, secure memory has the downside of having slower access speeds and being more expensive. As previously described, secret keys can be derived from PUF responses. Rather than store the keys in memory, they can instead be regenerated each time they are needed. This means that the only information that must be stored for each key is the challenge and whatever associated helper data required to generate it. This information is useless to an attacker that does not have access to the actual PUF and thus can be stored in normal unsecure memory.

4.2.3 PUF Design Methodologies

Silicon has proven to be a very popular medium for designing PUFs as researchers are able create designs based on transistor-level variations such as the propagation delay between gates [47] or the initial values found in memory when first powered on [141]. For self-contained devices such as IoT nodes, the implementation of these Silicon-based PUFs, especially ones based on propagation delay, would likely require the addition of specialized hardware or only be viable in certain applications. For example, the sensor node security protocol proposed in [11] uses a memory-based PUF created from the Static Random Access Memory (SRAM) found in commercial Bluetooth Low Energy (BLE) modules. Other researchers have begun exploring the feasibility of implementing a Dynamic Random Access Memory (DRAM)-based PUF in the existing memory of a Raspberry Pi B+ [29].

In addition to Silicon, there exists a wide range of components and materials which are suitable for PUF design [109]. The designs of Non-Silicon-based PUFs prove to be much more varied than normal silicon-based designs. Of particular interest are sensor-based PUFs as they are the category of PUF that our proposed design fits into. Comparatively little research exists on sensor-based PUFs. However, sensors and similar sorts of measurement devices are especially attractive for designing PUFs since their core functionality of measuring and reporting values can be directly incorporated into a PUF. Sensor PUF designs have been proposed based on a large range of components including microelectromechanical systems (MEMS)-based sensors [12, 159, 178], device touchscreens [143], photodiodes [139], solar cells [10], and piezoelectric sensors [94]. More information about these designs will be presented in Section V. For further information about other PUF designs, a number of existing comprehensive literature surveys are available. We point interested readers to any one of the following works: [46, 58, 190].

4.3 Proposed Design of Thermistor Temperature Sensor-Based PUF

A thermistor is a temperature sensing device whose resistance changes with temperature. The design of our proposed PUF uses the on the fact that variations introduced during the manufacturing process will cause individual thermistors to have different resistances at a

given temperature. These variations are what allow us to ultimately design a PUF capable of generating unique outputs.

In our proposed design we did not include the implementation of error correcting codes. Error correcting codes have already been proposed as a way to improve the reliability of responses by addressing faults such as bit-flip errors [26, 33, 40]. However, we wish to evaluate the baseline reliability of our proposed design. Adding error correction codes would obscure these values since the actual results would have been influenced by the codes. The addition of error correcting codes are thus a more relevant consideration for future work that would involve creating a production quality PUF from the proof of concept represented in this chapter.

4.3.1 Basic Circuit Diagram

The EK-TM4C123GXL model Tiva LaunchPad microcontroller we are using does not have a direct way to measure resistance. Instead, the board has a 12-bit analog-to-digital converter (ADC) capable of detecting voltages between 0 V and 3.3 V. For that reason we needed to create a circuit that would allow the changes in a given thermistor temperature sensor's resistance to manifest as voltage drops.

Our proposed solution is shown in Figure 4.1. The thermistors used in our design were NXP KTY81/220. Their operating parameters are shown in Table 4.1.

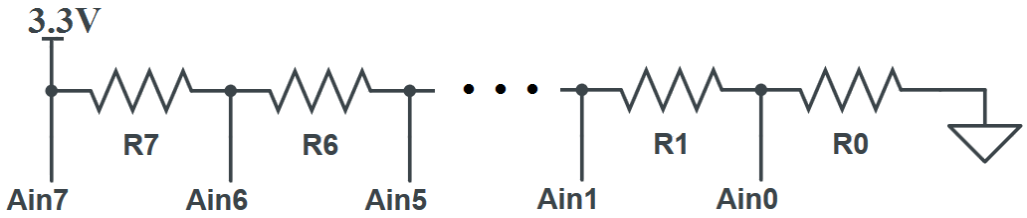


Figure 4.1: Proposed PUF Circuit Diagram (previously published in [98])

Table 4.1: Operating Parameters of NXP KRY81/220 Temperature Sensors [144] (previously published in [98])

Parameter	Value
Operating Temperature	-55°C to 150 °C
Typical Resistance @ -20 °C	1367 Ω
Typical Resistance @ 25 °C	2000 Ω
Typical Resistance @ 80 °C	2980 Ω

The entire circuit consists of 8 thermistor temperature sensors (here represented as resistors R) placed in series with a 3.3 V input voltage supplied by the microcontroller. A point before each thermistor is attached to an ADC input pin (A_{in}). The microcontroller is then able to take a voltage reading at each point and determine the voltage V_R across each thermistor R by finding the different between two surrounding points. For example, the voltage across thermistor $R5$ would be equal to the difference in readings between ADC inputs A_{in5}

and A_{in4}. The following equations show all the calculations that are made to determine the voltage across each thermistor:

$$\begin{aligned} V_{R7} &= A_{in7} - A_{in6} \\ V_{R6} &= A_{in6} - A_{in5} \\ V_{R5} &= A_{in5} - A_{in4} \\ V_{R4} &= A_{in4} - A_{in3} \\ V_{R3} &= A_{in3} - A_{in2} \\ V_{R2} &= A_{in2} - A_{in1} \\ V_{R1} &= A_{in1} - A_{in0} \\ V_{R0} &= A_{in0} \end{aligned} \tag{4.1}$$

Additionally, singular values read by the ADC can be noisy and slightly vary between readings. As a countermeasure, the final value for each ADC reading is actually the result of taking 100,000 readings and averaging the results.

4.3.2 Complete Architecture

Our proposed design requires 8 thermistor temperature sensors. Each sensor is connected to a microcontroller in the configuration shown in Figure 4.1. The onboard ADC is used to sample the voltage readings at each point and uses that data to ultimately derive a voltage drop across each thermistor. After this step is completed, an algorithm can be used to process the individual voltage data and construct a 128-bit response. One such example algorithm can be found in [94]. That algorithm generates a response by making a series of comparisons between total output readings for predetermined groups of a given component. That algorithm assumes that each component should have the same reading, and any differences are solely due to their intrinsic variations. This means that actions such as applying heat to some of the thermistors will result in unreliable readings. The end result is a PUF design that is directly based on thermistor temperature sensors. Figure 4.2 shows a picture of the fully constructed PUF.

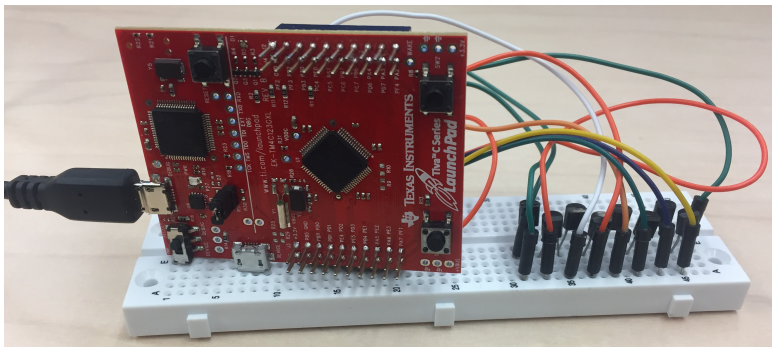


Figure 4.2: Prototype Implementation of Proposed Thermistor Based PUF

4.4 Testing Configuration and Results

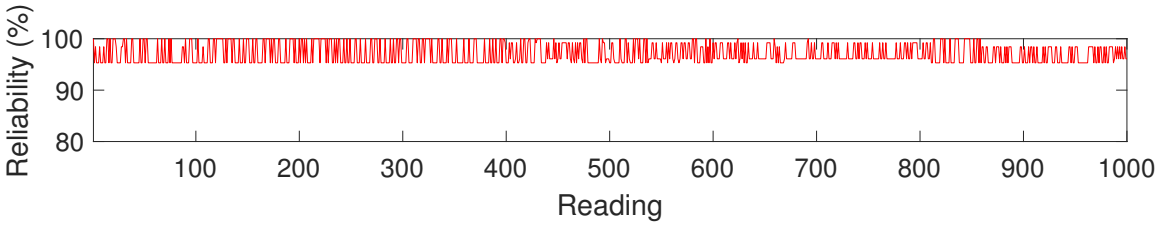
The responses generated from our proposed PUF design were tested to evaluate their reliability and uniformity (as originally described in [110]). In addition, we evaluated the uniqueness of the design by performing Monte Carlo simulations with 1000 simulated copies of the PUF.

4.4.1 Reliability Testing

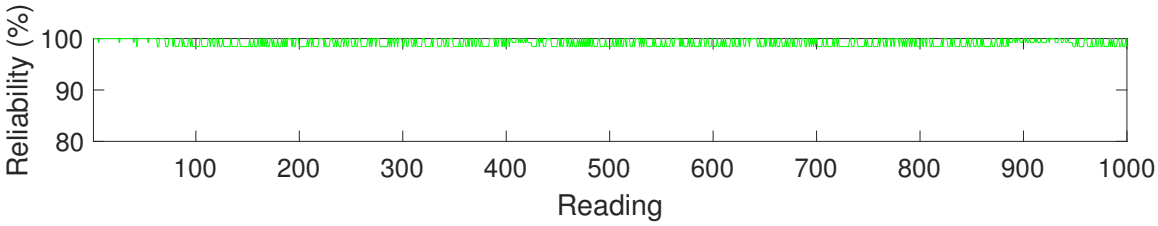
The reliability of a PUF is a measure of how often it will produce the correct response. The ideal reliability value of a PUF is 100%. This indicates that the PUF will never produce an erroneous response to a given challenge. The following equation (first described in [110]) is used to calculate the reliability for a n-bit response:

$$Reliability = 100\% - \frac{1}{m}\sum_{t=1}^{m}\frac{HD(R_i, R_{i,t}^{'})}{n} \times 100\% \tag{4.2}$$

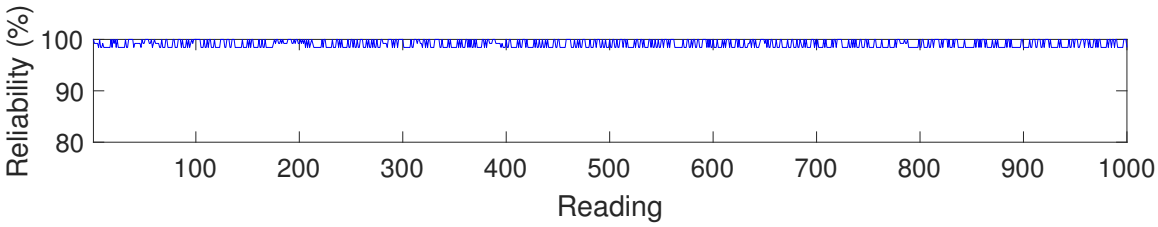
In this equation, R_i is a chosen reference response from PUF instance i. $R_i^{'}$ is a response generated under different environmental conditions. A total of m responses are collected with different environmental conditions. $HD(R_i, R_{i,t}^{'})$ is the hamming distance (HD) between the reference response (R_i) and the t-th generated response ($R_{i,t}^{'}$).



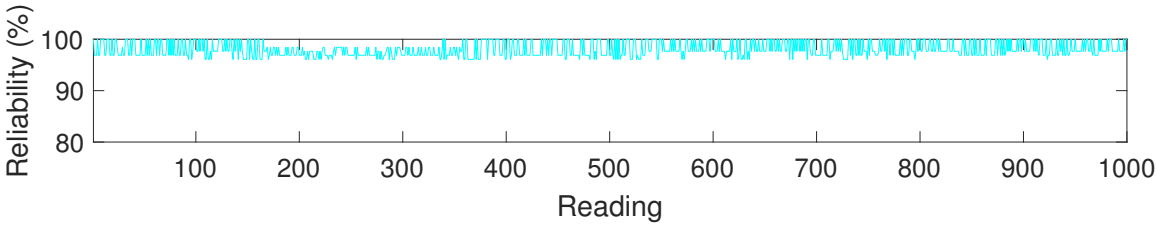
(**a**) PUF 1



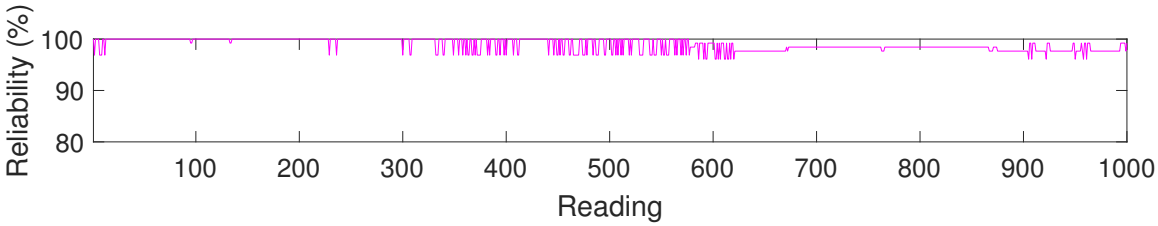
(**b**) PUF 2



(**c**) PUF 3



(**d**) PUF 4



(**e**) PUF 5

Figure 4.3: Reliability of PUFs Against Repeated Response Generation (previously published in [98])

For our initial reliability testing we took 1000 consecutive readings from 5 copies of our proposed PUF. The first response generated by each PUF was used as the reference response. All readings were taken in a lab space under normal room conditions. Figure 4.3 shows the graphs for the reliability values of the responses generated by each PUF. The graphs show that each PUF copy maintains a level of reliability that remains close to the ideal value of 100%. Table 4.2 contains the average reliability values for each copy

of the PUF. Among the five copies of the proposed PUF, PUF2 had the highest average reliability at 99.16% while PUF1 had the lowest at 97.09%. The overall combined average reliability for the tested copies was 98.46%.

Table 4.2: Average Reliability Values of Proposed PUF Instances when Generating 1000 Consecutive Responses (previously published in [98])

PUF1	**PUF2**	**PUF3**	**PUF4**	**PUF5**	**Total**
97.09%	99.16%	99.09%	98.08%	98.91%	98.46%

Temperature Reliability Testing

The next phase of reliability testing involved taking readings on each PUF over a range of -20 °C to 80 °C in increments of 5 °C. This was achieved by using the temperature chamber shown in Figure 4.4 and the graph of the results is shown in Figure 4.5.

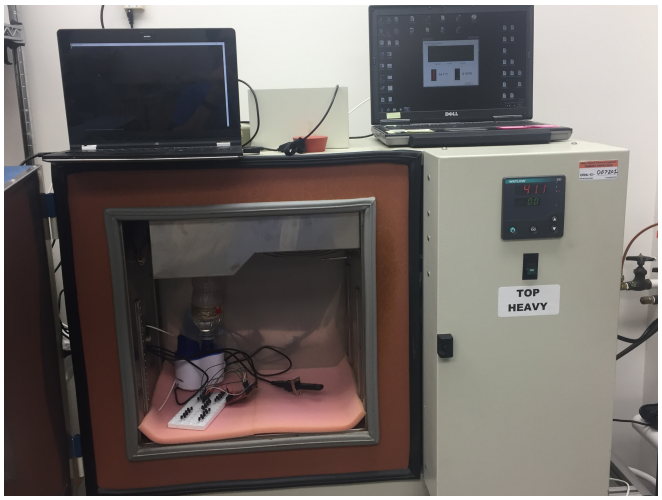


Figure 4.4: Testing Chamber (previously published in [98])

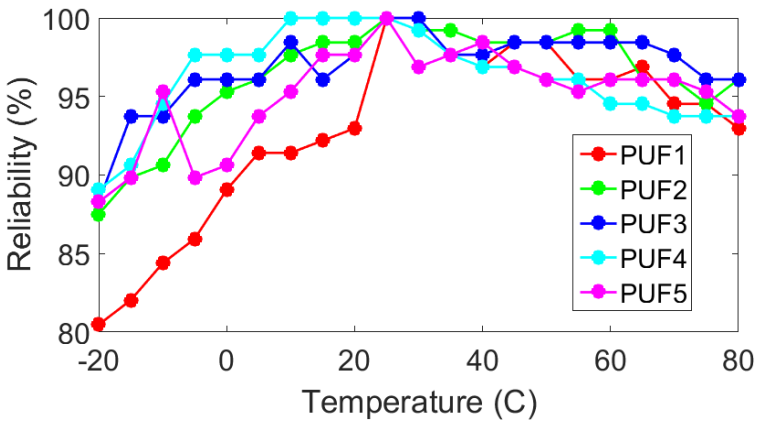


Figure 4.5: Reliability with Respect to Temperature. 25 °C was used as the reference value and the measured range was −20 °C to 80 °C in increments of 5 °C (previously published in [98])

25 °C was used as the reference temperature for determining the reliability values. This is why each copy of the PUF shows 100% reliability at 25 °C. The graph shows that the reliability values begin to fall off as the temperature moves away from the reference temperature of 25 °C. It is worth noting that PUF1 had a more pronounced decline than the other copies of the PUF did as the temperature moved towards −20°. This could be due to just random chance as the other 4 copies of the PUF remain relatively close together. In addition, PUF1 does not suffer a similarly drastic fall in reliability compared to the other copies of the PUF as the temperature approaches 80°. Even though PUF1's average reliability was a relatively respectable 92.97%, it was still the lowest average reliability among the tested PUFs. Table 4.3 shows the average reliability for each copy of the PUF. The overall total average reliability for the set was 95.49%.

Table 4.3: Average Reliability from −20 °C to 80 °C (previously published in [98])

PUF1	PUF2	PUF3	PUF4	PUF5	Total
92.97%	96.32%	96.84%	96.21%	95.09%	95.49%

Relative Humidity Reliability Testing

Reliability testing was also performed with respect to relative humidity. 30% relative humidity was used as the reference values and the relative humidity increased from 30% to 100% in increments of 10%. Figure 4.6 shows the reliability of the PUFs as the relative humidity increases from 30%. Overall, the PUFs seemed to be resistant to changes in relative humidity. Most copies did not show consistent drops in reliability until the relative humidity reached 80%. Table 4.4 shows the average reliability for each copy of the PUF. PUF1 once again demonstrated the lowest reliability of the test group with an average reliability of 95.70%. The overall total average reliability was 98.30%.

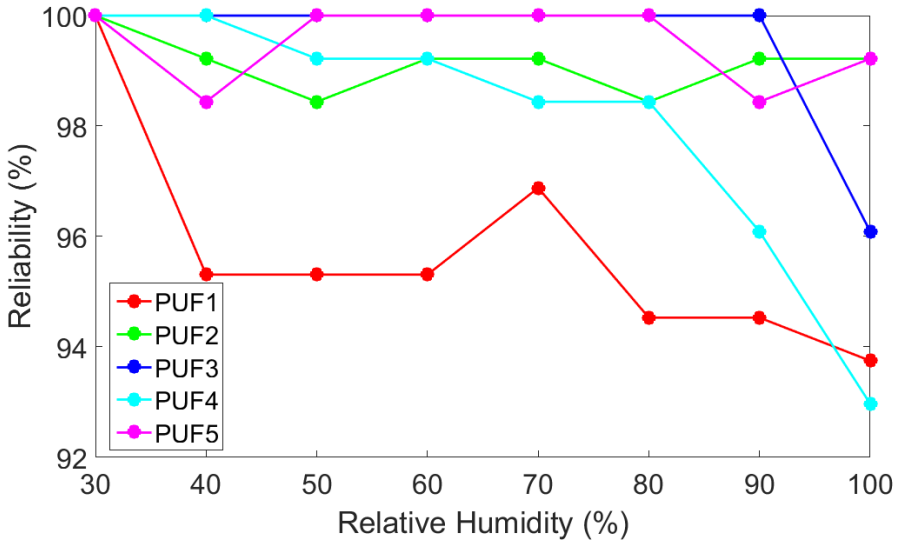


Figure 4.6: Reliability with Respect to Relative Humidity. 30% was used as the reference value and the measured range was 30% to 100% (previously published in [98])

Table 4.4: Average Reliability from 30% to 100% Relative Humidity (previously published in [98])

PUF1	**PUF2**	**PUF3**	**PUF4**	**PUF5**	**Total**
95.70%	99.12%	99.12%	98.05%	99.51%	98.30%

4.4.2 Uniformity Testing

The uniformity of a PUF describes how "balanced" its responses are, i.e., what is the prevalence of 1's vs. 0's in the bits of the responses. Ideally, there will be an equal number of 1's and 0's to maximize the difficulty for an attacker trying to guess the value of a given bit. This ideal scenario is represented by a uniformity value of 50%. The following equation (first described in [110]) is used to calculate the uniformity of a n-bit response:

$$Uniformity = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} \times 100\% \tag{4.3}$$

In the above equation, $r_{i,l}$ represents the l-th bit of a n-bit long response generated by PUF instance i. In order to obtain a general uniformity of PUF we averaged together all the readings for a given test. Table 4.5 shows the average uniformity value for each copy of the proposed PUF across each of the areas of testing (1000 consecutive responses, temperature, and humidity). The overall average uniformity values for the different tests were 50.22%, 49.34%, and 47.91%, respectively. On average, the uniformity values were very close to the ideal value of 50%.

Table 4.5: Average Uniformity Values of Proposed PUF Instances (previously published in [98])

	PUF1	**PUF2**	**PUF3**	**PUF4**	**PUF5**	**Total**
Consecutive	49.66%	49.96%	50.05%	49.48%	51.94%	50.22%
Temperature	48.59%	48.21%	49.52%	48.92%	51.45%	49.34%
Humidity	47.46%	46.58%	49.51%	47.85%	48.14%	47.91%

4.4.3 Uniqueness Testing

As described in [110], the uniqueness of a PUF represents the ability to distinguish one particular instance of a PUF from a group of PUFs of the same type. The ideal uniqueness value is 50%. The following equation is used to calculate uniqueness:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \tag{4.4}$$

The above equation determines the average hamming distance (HD) among k total PUFs. R_i and R_j represent n-bit responses produced by PUFs i and j, respectively where $i \neq j$.

The common method for evaluating the uniqueness property of a PUF is by performing Monte Carlo simulations as this allows many unique copies to be generated. For our simulations we created 1000 simulated copies of the PUF. We first created a normal distribution of resistors using the manufacturer specified resistances at 25 °C [144]: minimum of 1960 Ω, maximum of 2040 Ω, and typical of 2000 Ω. Each simulated instance was created by randomly choosing 8 resistors from the distribution. The uniqueness was determined to be 49.89%.

4.5 Comparison to Existing Designs

It should be noted that other PUF designs which are effectively based on measuring differences in resistance values have been proposed. Those designs are based on materials such as magnetoresistive RAM (MRAM)[37], memristors [138, 166], and on-chip transistors [76] and metal wires [75, 76]. These designs share a common theme with our proposed thermistor PUF of using unique resistances to produce a response. However, we do not feel this is strong enough of a justification to include these designs in direct comparisons that we will do with other sensor PUF designs. The main reason is that one of the goals in creating sensor PUFs is that theoretically a device that already contains the requisite number of sensors could function as a PUF without needing to add any additional hardware. Much like Silicon PUFs, these resistance-measuring PUFs would have to be specifically added to the target device. Furthermore, the resistances of thermistors are designed to change with temperature and can therefore be more sensitive than the components in other designs. Variations in physical properties due to temperature are not intended to be the core

operating mechanic of those designs (e.g., allowing thermistors to measure temperature). Sensors on the other hand are generally designed to change one of their physical properties in a significant and predictable way as a direct response to the environmental condition they are monitoring. That same physical property also serves as the basis for creating a sensor PUF from a given sensor. It is for these reasons the focus of our comparisons will be PUF designs that are based on sensors.

Certain difficulties were encountered when attempting to compare the results of our proposed thermistor PUF to existing sensor PUF designs. Unfortunately, sensor PUFs are less popular than Silicon-based PUFs and thus there is comparatively little directly applicable existing research for which we can compare our work. This is further exacerbated by the fact that works that have proposed sensor-based designs do not tend to include performance metrics that can be directly compared with our results. Among the existing sensor-based PUF designs, we are only able to make a direct comparison of performance metrics with the piezo sensor-based design [94]. Comparisons to other designs will solely focus on the functional aspects of the PUF designs.

The devices that we will specifically highlight are microelectromechanical systems (MEMs)-based sensors [12, 159, 178], device touchscreens [143], photodiodes [139], solar cells [10], and piezo sensors [94]. These designs have certain drawbacks that could hinder their adoption by cyber-physical systems. The piezo sensor PUF [94] requires a sinusoidal input source which is not always readily available certain devices. Additionally, piezo sensors cannot be considered to be prevalent as thermistor temperature sensors since vibration sensing is less common when compared to temperature sensing. The MEMs gyroscope designs [12, 178] generate responses based on the output of a MEMs gyroscope. The major concern would be how easily a given gyroscope orientation could be reproduced by a user. A different MEMs-based approach is a ring oscillator (RO) PUF design in which the ring oscillators are constructed from pressure sensing MEMs relays [159]. This design is costly as it requires a separate RO for each bit in the response in addition to bias generation circuitry to control the relays. The touchscreen design [143] is subject to the same type of concern. The design generates a response based on a user's ability to trace a specified pattern on the screen. There should be a certain amount of variance in results every time a user attempts to replicate the same fine movements that would be used to trace a specified pattern. The photodiode-based design [139] is subject to a sort of chicken and egg problem where its design actually requires a conventional PUF to operate. Lastly, the solar cell work [10] shows that solar cells could potentially be used as a PUF, but stops short of proposing a complete design. Table 4.6 contains a summary of the drawbacks of various sensor PUF designs.

Our proposed design does not suffer from any of the previously mentioned drawbacks that are present in existing designs. One potential concern is the number of thermistors required to implement our proposed design will not always be present in a given cyber-physical device or system. However, some areas such as certain industrial applications [50, 71] which make use of redundant temperature sensors could be especially suitable thanks to the larger than normal number of temperature sensors.

In terms of actual performance metrics, we were only able to make direct comparisons with the reliability and uniformity results between our proposed design and those from the piezo sensor-based PUF [94]. Uniqueness values were not reported. Table 4.7 contains

these values for both our proposed design and the previous piezo sensor work. The average reliability and uniformity across three copies of the piezo PUF was calculated to be 96.07% and 47.24%, respectively. Our proposed design had an average reliability of 98.46% and an average uniformity of 50.22%.

This improvement could be attributed to a couple of factors. The first possibility is the circuit used by our proposed PUF could be more conducive to producing consistent responses. The piezo design required using an ADC to sample AC waveforms which could introduce noise into the measurements. The fact that our proposed design samples what should be steady DC voltages means that the overall sampling process is more straightforward and thus more consistent. A second possible contributing factor is some unspecified aspect of the physical properties of thermistor temperature sensors could simply make them better suited than piezo sensors for constructing PUFs.

Table 4.6: PUF Comparison (previously published in [98])

PUF	**Description**	**Drawback**
Piezo [94]	Compares summations of voltage drops across groups of piezo sensors	Requires an additional AC input voltage. Limited applications compared to proposed design.
MEMs Gyro [12, 178]	Responses are derived from the output of a MEMs gyroscope	Concerns about being able to repeatedly produce a desired CRP.
MEMs Pressure [159]	Ring Oscillator (RO) design using pressure sensitive MEMs relays.	Significant overhead due to additional circuitry.
Touchscreen [143]	A user traces a specified pattern displayed on the touchscreen	Concerns about being able to repeatedly produce a desired CRP.
Photodiode [139]	Compares summation of sensor groups based on the output of a PUF	Correct operation requires an existing conventional PUF.
Solar Cells [10]	Testing results show that solar cells produce unique voltages for the same light source	Complete design not proposed.
Proposed Design	Uses microcontroller to compare readings from groups of thermistor temperature sensors to generate a weak response.	Requires more thermistor temperature sensors than may already exist in certain systems.

Table 4.7: PUF Comparison (previously published in [98])

	Piezo [94]	**Proposed**
Uniformity	47.24%	**50.22**%
Reliability	96.07%	**98.46**%

A direct comparison of reliability with respect to temperature is complicated by the testing method employed for the piezo PUF. Both our proposed PUF and the piezo PUF used 25 °C as a reference temperature. However, two different chambers were used to test ranges of -20 °C to 0 °C and 25 °C and 80 °C with the range of 0 °C to 25 °C being extrapolated. This prevents a direct comparison in terms of average reliability values. What can be noted is the reliability for the piezo PUF faces a much sharper drop in reliability (below roughly 88%) than any of the thermistor PUFs in which the lowest recorded reliability was 92.97% at 80 °C for PUF1. Additionally, for the range of -20 °C to 0 °C the piezo PUF had its reliability generally drop as the temperature approached 0 °C. Its reliability at -20 °C was better than all the tested copies of our proposed thermistor temperature sensor PUF. However, its reliability at 0 °C was worse than any of the copies of our proposed thermistor temperature sensor PUF.

4.6 Discussion & Conclusions

In this chapter, we have proposed a novel PUF design for use in cyber-physical systems by using thermistors which are components commonly found within the field. The actual design uses a microcontroller to compare the summed voltage outputs across predetermined groups of thermistor temperature sensors to generate a weak response. Monte Carlo simulations produced a uniqueness value of 49.89% which is very close to the ideal value of 50%. Our proposed design was shown to have improved overall reliability and with regards to changes in temperature when compared to the existing design based on piezo sensors [94]. Additional reliability testing with respect to relative humidity appeared to show that the proposed design is relatively unaffected by humidity values less than 80%. As a future work, the addition of error correcting codes could help improve the reliability values of the base design.

It is worth noting that this design should be treated as a proof of concept and not a fully realized security solution. The main goal in creating this device was to conduct a preliminary exploration to determine if thermistor temperature sensors are a viable option for PUF creation when compared to existing sensor-based PUF designs. The prototypes we created for testing purposes were meant to only address this question of viability. The prototypes are vulnerable to physical attacks such as an attacker manually measuring the voltage drops across each thermistor and then creating a model of the PUF. Other researchers have already explored mitigation methods such as implementing tamper-resistance [64, 103] and providing protection from side channel attacks [77, 127, 158]. Exploring the integration of existing solutions or devising new concepts are outside of the scope of this chapter and should instead be considered to be avenues for future work when designing a full-scale

production quality implementation.

Chapter 5

Exploration of Solar Cell Materials for Developing Novel Physically Unclonable Functions in Cyber-Physical Systems

Energy-efficiency is a major concern of the Internet of Things (IoT) devices found within cyber-physical systems. Solar cells are one common option for providing a source of power to these devices. A PUF designed using solar cells has the potential to not only serve as a source of power generation, but also provide security. In this chapter, we propose a novel PUF architecture based on solar cells. The proposed design utilizes a microcontroller to read the open-circuit voltages (V_{oc}) of a selection of solar cells and generate an associated response. The proposed design was implemented using amorphous silicon solar cells, monocrystalline solar cells, and polycrystalline solar cells. Furthermore, we evaluated the reliability and uniformity for each type of PUF against variations in temperature and variations in light intensity. We also performed uniqueness testing on monocrystalline silicon solar cells via Monte Carlo simulations.

The rest of this chapter is organized as follows: Section I introduces the motivation for the construction of PUFs from solar cells. In Section II we provide information on PUFs and solar cells. Section III covers our proposed design methodology including the underlying architecture. Section IV presents the implementation details of our proposed design including information on each type of solar cell that was used. Section V describes the various types of testing and provides the results. Section VI provides a more in-depth discussion of the results including an analysis of each PUF's relative performance. Section VII gives conclusions and offers avenues for potential future work. Material from this chapter was previously published as "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd".

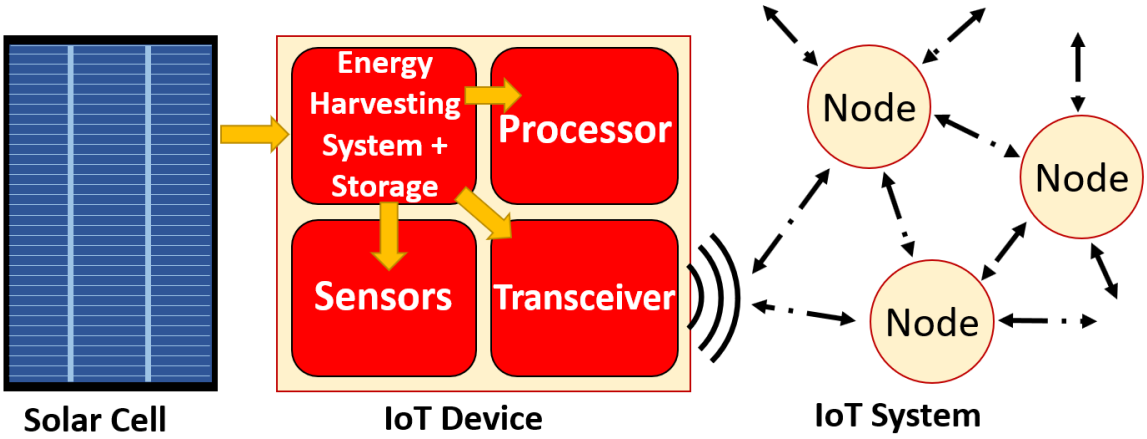


Figure 5.1: Example of integration of solar cells within an IoT system. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

5.1 Introduction

The Internet of Things (IoT) consists of a network of Internet connected devices that have the ability to transmit information as part of their use in intelligent applications [13]. These cyber-physical systems offer quality of life improvements by allowing physical objects to be directly integrated with the digital world. In general, IoT devices tend to be small and contain a limited battery supply. As a result, these devices are typically subjected to strict constraints on their power consumption and available hardware resources [2].

Energy harvesting is one technique that shows some promise as a way to improve energy efficiency [84]. Among the various energy harvesting technologies, the use of solar cells to harvest solar energy provides the highest power density. They are therefore an ideal choice for powering IoT devices as shown in Fig. 5.1 [104]. Highly efficient solar cells could even prove to be a viable option for indefinitely powering an IoT device without requiring a battery replacement. One such example is the Gallium Arsenide (GaAs) solar cell developed by Alta Devices which has an efficiency of 28.8% [186]. The integration of cells such as these could drastically increase the amount of time an IoT device can operate without needing a battery replacement, or even completely eliminate the need to replace the battery.

IoT devices also face major challenges in the form of security. Two major areas of concern are authentication and access control [112],[74]. The non-volatile memories used to store secret keys have been shown to be vulnerable to active attacks [7], [91]. Furthermore, it may be too expensive in terms of cost and energy to introduce high level security through the addition of tamper resistant circuitry.

Physically Unclonable Functions (PUFs) have drawn interest as a hardware security primitive that could be more specially suited for integration in resource constrained devices. PUFs have been shown to address a variety of security concerns such as IC piracy, counterfeiting, etc. [155]. They can be used in cyber-physical security and IoT devices as

a major component in protocols for secure authentication and key management [5] [160] [124].

5.1.1 Motivation

PUFs are a type of circuit which utilize intrinsic variations introduced during the manufacturing process to create devices that are unique and unclonable. PUFs are commonly based on CMOS ICs which require dedicated hardware to function correctly. This dedicated hardware results in additional costs in terms of hardware and power consumption. In recent years, IoT devices have begun incorporating solar cells for tasks such as power generation and sensing. The goal of this work is the creation of a PUF whose integration with IoT devices would incur a minimal cost in the form of additional hardware. The use of solar cells provides the potential for a PUF that can be used as a source of power in addition to security. Solar cells are already a common power source in many remote applications and locations such as satellites, roadside displays, building rooftops, etc. By finding a way to leverage these existing components one could add security features without having to add any additional hardware. The cells' main purpose would be powering the device, but they would have the added advantage of also being able to be used as a method of providing security features. Our proposed design should be considered an energy harvesting based PUF and to the best of our knowledge is the first work to perform extensive testing on different types of solar cells to evaluate their viability for creating PUFs for cyber-physical systems.

5.2 Background

5.2.1 Physically Unclonable Functions

PUFs are a type of circuit which utilize intrinsic variations introduced during the manufacturing process to create devices that are unique and unclonable. PUFs could serve as a solution to security problems such as IC piracy, counterfeiting, etc. [155] as their inability to be cloned would allow for the unambiguous identification of valid devices. Furthermore, they can be used in cyber-physical security and IoT devices as a major component in protocols for secure authentication and key management [5] [160] [124]. Silicon based PUFs such as arbiter PUF [101], Ring Oscillator (RO) [155], SRAM PUF [59] have proven to be very popular. These types of PUFs utilize transistor level variations to generate unique responses.

5.2.2 Solar Cells

Solar cells are known for their ability to convert energy from a light source into electricity with a relatively high conversion efficiency. They can serve as a nearly permanent source of power with low operating costs while also being virtually free of pollution. Solar cells generate both voltage and current. This is accomplished by using absorbed light to raise electrons to a higher energy state which can then be transported into an external circuit. The separation of photo-generated electrons and holes is achieved through the use of p-n

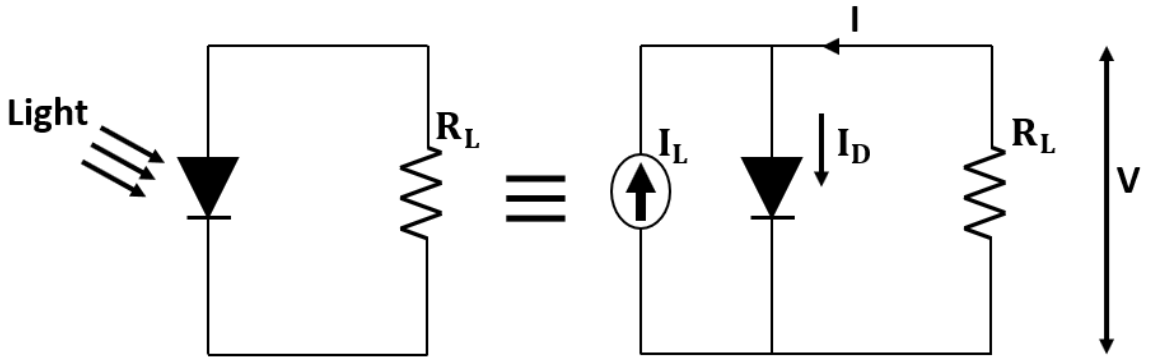


Figure 5.2: Solar cell equivalent circuit. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

junctions constructed from semiconductor and inorganic-organic materials (Fig. 5.2). It has been demonstrated that the incorporation of solar cells can improve the energy efficiency of IoT devices [183] and by extension improve the energy efficiency of cyber-physical systems which contain IoT devices. Other researchers have performed preliminary investigations on designing PUFs based on solar cells [139] [10] [89]. However, either those explorations weren't fully formed or the responses produced by those designs were directly tied to the current light intensity. This should prove a major hindrance towards their implementation in IoT devices as the light intensity of the operating environment is not typically something that can be easily modulated.

5.3 Methodology

The first step in designing a PUF based on solar cells is to choose an appropriate electrical parameter of the solar cells that will serve as the basis for ultimately generating a response. We believe that choosing a parameter that has a predictable relationship with changes in environment operating conditions will ultimately result in a more reliable PUF. In this section, we discuss our chosen solar cell electrical parameter that forms the basis of our proposed PUF.

5.3.1 Parameters to Design Solar Cell PUF

Perhaps the two most important electrical parameters of solar cells are their open circuit voltage (V_{oc}) and their short circuit current (I_{sc}). Both of these quantities are easily measurable and are fundamentally related through the current-voltage (I-V) equation of a solar cell [183]. That equation can be seen below:

$$I = I_0[exp(\frac{qV}{\eta kT}) - 1] - I_L \tag{5.1}$$

where I_0 is the reverse saturation current, q is the electron's charge, η is the diode ideality factor, k is the Boltzmann constant, T is the temperature, and I_L is the light generated current. V_{oc} is the maximum voltage that can be generated by a solar cell and it occurs when the current $I = 0$. Similarly, I_{sc} is the largest current that the cell can produce and is directly dependent on the spectrum of the incident light, i.e. the number of photons and the quantum efficiency of the solar cell. The previous equation can be rewritten as follows:

$$V_{oc} = \frac{\eta kT}{q} ln(\frac{I_L}{I_0} + 1) \tag{5.2}$$

This equation can be used to directly relate I_{sc} and V_{oc}, because in an ideal situation I_{sc} is equal to I_L. As demonstrated in the above equation, the value of V_{oc} for a solar cell is the direct result of the light generated current I_L and the reverse saturation current I_0. However, I_0 actually has a much greater influence on the value of V_{oc} as I_L tends to have only small variations in value while I_0 can actually vary by orders of magnitude. This is because I_0 is itself dependent on many solar cell characteristics such as electron-hole recombination lifetimes, interface state density, defects and impurities, etc. As a result, this value can vary wildly even among cells that are otherwise "identically produced". This high degree of entropy actually makes it an ideal candidate to serve as the basis for designing a PUF. Unfortunately, the reverse saturation current is not a quantity that can be as easily measured by a PUF during normal operation and thus other parameters should be considered.

As previously explained and shown in Equation (2), I_0 is the only parameter that contributes to V_{oc} that is also known to show orders of magnitudes in variations. This means that any variations in I_0 should likewise manifest in V_{oc} which is far easier to measure than I_0. It is known that solar cells which should be otherwise identical, such as produced from the same batch, will actually display variations in their V_{oc} values due to intrinsic variations introduced during the manufacturing process. Furthermore, the V_{oc} values of solar cells are known to respond in a predictable manner when subjected to changes in both temperature (linear relationship) and light intensity (logarithmic relationship). We believe this predictability will result in a PUF that is able to generate reliable data in various operating conditions. It is for these reasons that we selected the open-circuit voltage (V_{oc}) as the solar cell parameter on which to base our proposed PUF design.

Solar cells have been constructed from a myriad of different elements. However, silicon has proven to be a very popular choice and therefore we chose to use silicon solar cells with our proposed solar cell based PUF design. Furthermore, we evaluated our proposed design using multiple types of silicon solar cells to determine their viability in different environmental conditions. The specific types of silicon solar cells used were Panasonic AM-1417CA amorphous silicon solar cells ($V_{oc} = 2.4V$), IXYS KXOB22-12X1F monocrystalline solar cells ($V_{oc} = 630mV$), and AOSHIKE micro solar panel polycrystalline silicon solar cells ($V_{out} = 2V$).

5.3.2 Proposed PUF Architecture

Through the photovoltaic effect, solar cells generate a voltage when they are hit by photons. These output voltages will actually vary between solar cells due to intrinsic variations introduced during the manufacturing process. Our proposed design uses a microcontroller

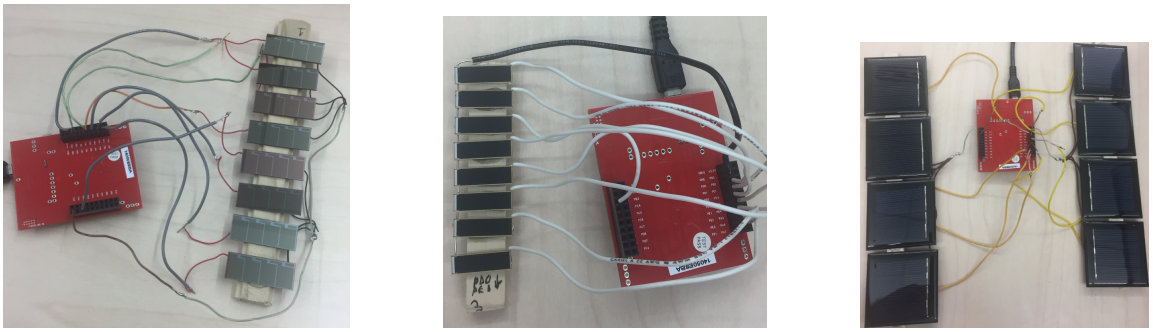
with an ADC to first capture these output voltages and convert them to digital values. The PUF uses these values to generate a 128 bit response by comparing the voltages in a pre-determined pattern. Each bit in the generated response is a direct result of a comparison made between the output voltages from two different groups of solar cells. 128 bits was chosen since it is a commonly used response size and larger response sizes could require more solar cells to implement. Further explanation on the actual hardware and software portions of our proposed design can be found below in Section IV.

5.4 Implementation

In this section, we present actual prototypes of our proposed solar cell based PUF. This section will highlight the various hardware (solar cells and microcontroller) and software components of our proposed design.

5.4.1 Hardware Components of Proposed PUF

Each prototype consists of 8 solar cells connected to ADC input pins on a microcontroller. A personal computer (PC) is used to communicate with the PUFs. In our implementations, we have used a selection of solar cells. The three types of solar cells were monocrystalline silicon, polycrystalline silicon, and amorphous silicon. A copy of each type of PUF is shown in Figure 5.3.



(**a**) Amorphous silicon PUF (**b**) Monocrystalline silicon PUF (**c**) Polycrystalline silicon PUF

Figure 5.3: Prototype solar cell based PUFs. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

Amorphous Silicon Solar Cells

The amorphous cells used in our design were Panasonic AM-1417CA amorphous silicon solar cells [130]. Amorphous silicon cells do not have the regular atomic arrangements that are present in crystal silicon cells. This irregular atomic arrangement allows for more light

absorption and thus certain types of amorphous silicon cells can be produced that have a film thicknesses of less than 1 μm [130]. The specific model we tested was designed for use in indoor applications such as wireless sensor networks, RF remote controls, battery chargers, etc. They have an open-circuit voltage (V_{oc}) of 2.4V with a max power of 18.75 μW. Table 5.1 provides a complete list of the electrical parameters of the amorphous silicon solar cells.

Table 5.1: Electrical Parameters of the Amorphous Silicon Solar Cells Used in our Experiments

Cell parameters	Typical ratings
Open-circuit voltage	2.4V
Short circuit current	13.5 μA
Max. power	18.75 μW
Voltage at max. power point	1.5V
Current at max. power point	12.5 μA

Monocrystalline Silicon Solar Cells

The monocrystalline cells used were the IXYS KXOB22-12X1F monocrystalline solar cells [70] which are used for various battery operated consumer products such as mobile phones, cameras, MP3 players, etc. These solar cells also have applications in IoT based devices such as wireless sensors, RFID tags, etc. These solar cells have very good response over a wide wavelength range and therefore can be used in a variety of indoor and outdoor applications. They have an open-circuit voltage (V_{oc}) of 630 mV with an efficiency of 22%. Table 5.2 provides a complete list of the electrical parameters of the monocrystalline silicon solar cells.

Table 5.2: Electrical Parameters of the Monocrystalline Silicon Solar Cells Used in our Experiments

Cell parameters	Typical ratings
Open-circuit voltage	630 mV
Short circuit current density	42.4 mA/cm^2
Max. peak power	18.6 mW/cm^2
Voltage at max. power point	501 mV
Fill factor	$\geq$ 70%
Solar cell efficiency	22%

Polycrystalline Silicon Solar Cells

Polycrystalline silicon solar cells are made of multiple silicon crystals. This differs from monocrystalline silicon solar cells where the entire cell is comprised of a single silicon

crystal. Polycrystalline tends to have a cheaper manufacturing process than monocrystalline silicon. However, the existance of multiple crystals means they also tend to be less efficient. The polycrystalline silicon solar cells were AOSHIKE micro solar panel polycrystalline silicon solar cells. We were not able to find a data sheet for the cells. The information provided by their Amazon listing rates their peformance as an output voltage (V_{out}) of 2V and a current of 130 mA. They are a suitable power source in applications such as low-power electrical appliances, small motors, solar water pumps, lighting, etc. [9].

Microcontroller

Our proposed design requires using an ADC and a microcontroller to measure the solar cell output voltages. These voltage values are compared in a pre-determined pattern to generate a 128-bit response. Our example implementations (Figure 5.3) use a Tiva TM4C123GH6PM. We chose these as they have already been included in multiple applications such as network appliances and switches, remote monitoring, factory automation, etc. [67]. For testing purposes we use a PC to send challenges and receive responses via UART.

The ADC within that board is 12 bits. It should not necessarily be viewed as a requirement that other implementations must also use a 12-bit ADC. Our testing results from the next section will show that it is sufficient for creating a PUF. It is worth noting that changes in ADC resolution could cause differences in the results. At a minimum, other implementations should use an ADC with a high enough resolution that it is able to detect the voltage variations between each solar cell. Realistically a PUF is not going to have its ADC replaced so there is not a major concern on how changing the resolution would change the response. This would effectively be replacing the PUF in which case it is no longer expected to produce the same response. The exact effect, if any, that changing the ADC's resolution would have on the PUFs, we consider that to be outside the scope of the work presented in this chapter.

5.4.2 Software Components of Proposed PUF

The software running on the microcontroller is responsible for actually generating the PUF's response. The microcontroller must sample each connected solar cell through its ADC and then generate a response. For the actual comparison algorithm used to generate the response, we used the one described in [92]. Our implementation required accounting for the inherent noise in the ADC. This was done by averaging 16,000 readings per cell. The sampling rate was 125,000 samples per second and microcontroller speed was 20MHz. A PC was used to communicate via UART with the PUFs. This was mostly for testing purposes as it allowed us to easily record the generated responses.

5.5 Testing and Results

Reliability testing and uniformity testing were performed on three copies of each type of solar cell PUF. Both metrics were evaluated with respect to changes in temperature and

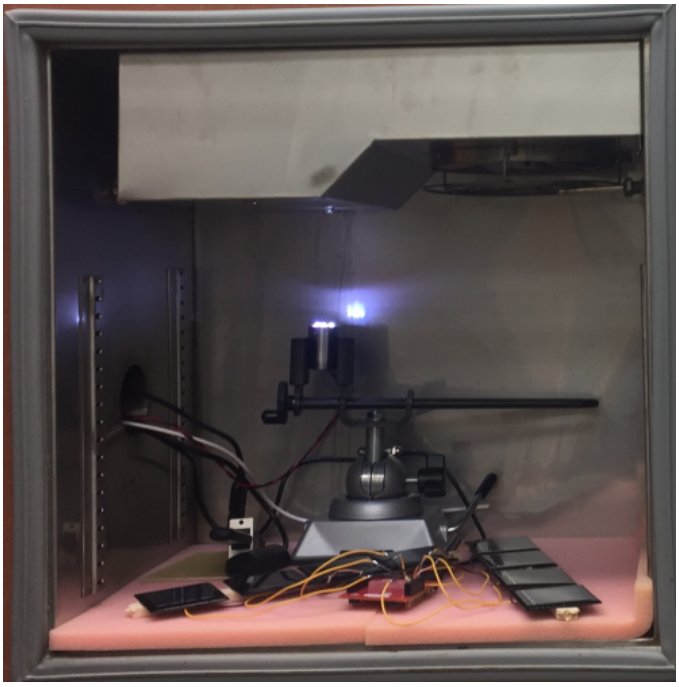


Figure 5.4: Temperature testing chamber. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

changes in light intensity.

5.5.1 Testing with Respect to Temperature

A temperature chamber was used to evaluate the reliability of the generated responses over a range of temperatures from -20°C to 80°C. 25°C was used as the reference temperature for reliability testing and measurements were taken in increments of 5°C. A "LED-LENSER® V6 7732" was used to provide a constant source of light within the sealed testing chamber. The complete testing setup is shown in Figure 5.4.

Reliability Testing

The reliability of a PUF is a measure of how well it can reproduce a given response with respect to changes in a specified environment condition such as temperature. The reliability of a n-bit response can be calculated by the following equation:

$$Reliability = 100\% - \frac{1}{k}\sum_{i=1}^{k}\frac{HD(R_i, R'_{i,t})}{n} \times \% \tag{5.3}$$

where HD denotes the hamming distance between a reference response R_i from PUF i and a separate response $R'_{i,t}$ from PUF i that has been generated under different environmental conditions. A total of k n-bit responses are generated under different environmental

conditions in order to calculate the average hamming distance. The ideal reliability value is 100% which indicates that the PUF will always generate the correct response regardless of changes in its operating environment.

Figure 5.5 shows the reliability values of the amorphous silicon PUFs and Table 5.3 shows the average reliability values for each copy of the PUF across the measured temperature range. The first copy of the PUF (PUF1) showed a remarkable consistency that manifested as an average reliability of 96.39%. However, this was an outlier as the other copies of the PUF did not fare nearly as well. There were pronounced drops in reliability as the temperature moved farther away from the reference value of 25°C and as a result overall average reliability was only 84.41%.

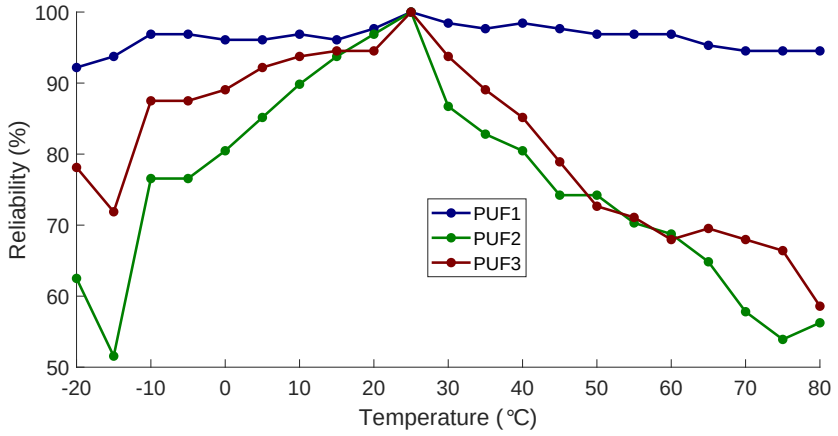


Figure 5.5: Amorphous silicon reliability with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

Figure 5.6 shows the reliability values for each of the monocrystalline silicon PUFs and Table 5.3 contains the average reliability for each copy of the PUF. The reliability of each copy of the PUF exhibited consistent changes as the temperature deviated from the reference value of 25°C and resulted in an overall average of 88.87%.

Figure 5.7 shows the reliability values for each of the polycrystalline silicon PUFs and Table 5.3 contains the average reliability for each copy of the PUF. The reliability values of the second and third copies of the PUF (PUF2 and PUF3) showed consistent behavior across the range of temperatures measured. PUF1 displayed similar reliability values for temperatures up to 50°C. Beyond that temperature the device demonstrated a notable drop in reliability that did not occur in the other copies of the PUF. Despite this, the PUFs still produced an overall average reliability of 91.20%.

The polycrystalline silicon PUFs displayed the highest average reliability at 91.20% while the amorphous silicon PUFs displayed the lowest average reliability at 84.41%. The monocrystalline PUF displayed the most consistency as it was the only type to have the reliability values of each copy fall within 2.5 percentage points of each other.

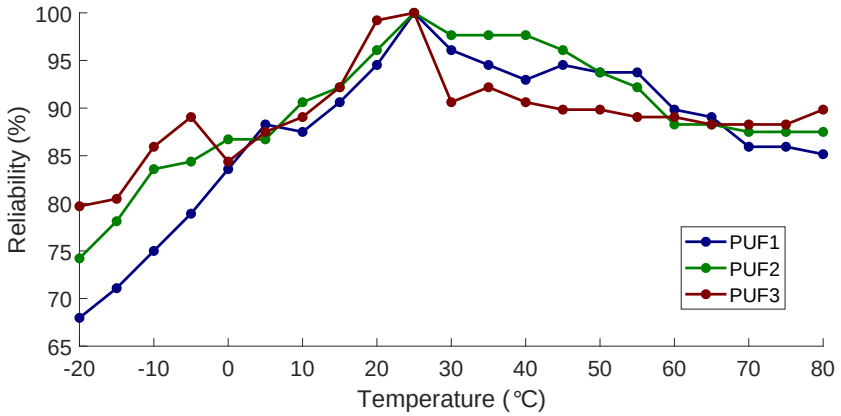


Figure 5.6: Monocrystalline silicon reliability with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

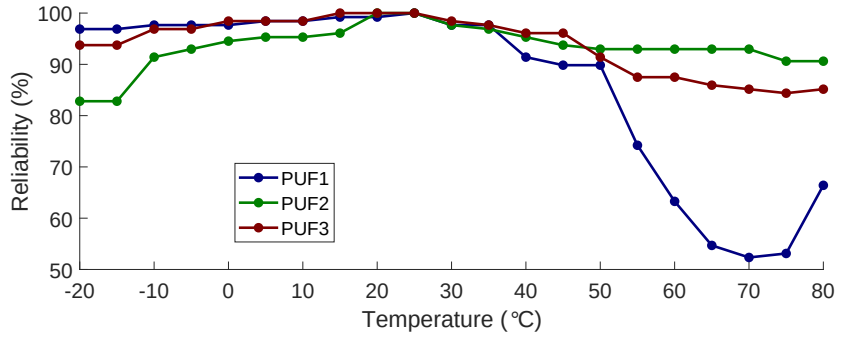


Figure 5.7: Polycrystalline silicon reliability with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

Uniformity Testing

The uniformity is a measure of how balanced a PUF's response is in terms of the number of 1's and 0's. The ideal uniformity value is 50% which denotes that there are an equal number of 1's and 0's. Uniformity can be calculated by the following equation:

$$Uniformity = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} \times 100\% \tag{5.4}$$

where $r_{i,l}$ represents the l-th bit of response from PUF instance i.

Figure 5.8 shows the uniformity values of the amorphous silicon PUFs and Table 5.4 shows the average uniformity values for each copy of the PUF across the measured temperature range. The three amorphous silicon PUFs produced a combined average uniformity of 49.34% across the tested temperature range.

Table 5.3: Average Reliability with Respect to Temperature

	PUF1	PUF2	PUF3	Overall
Amorphous	96.39 %	75.41 %	81.44 %	84.41 %
Monocrystalline	87.57 %	89.84 %	89.21 %	88.87 %
Polycrystalline	86.31 %	93.38 %	93.90 %	91.20 %

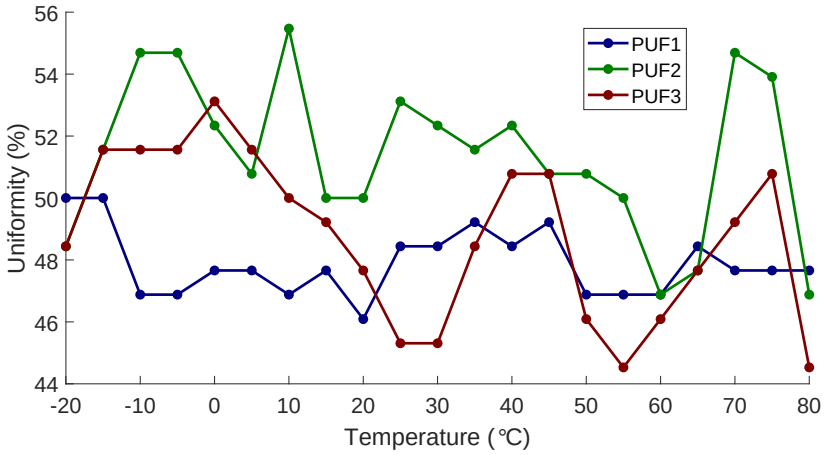


Figure 5.8: Amorphous silicon uniformity with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

Figure 5.9 shows the uniformity values for each of the monocrystalline silicon PUFs and Table 5.4 contains the average uniformity for each copy of the PUF. Overall, the monocrystalline silicon PUFs had a combined average uniformity of 49.47%.

Figure 5.10 shows the uniformity values for each of the polycrystalline silicon PUFs and Table 5.4 contains the average uniformity for each copy of the PUF. Overall, the polycrystalline silicon PUFs had a combined average uniformity of 51.26%

Despite changes in temperature, the uniformity values for each response generated by the different PUFs remained near the ideal value of 50%. The monocrystalline silicon PUFs had the average uniformity closest to the ideal value at 49.47%. Even though the polycrystalline silicon PUFs had the worst overall average uniformity, their overall average value of 51.26% was still close to the ideal value.

Table 5.4: Average Uniformity with Respect to Temperature

	PUF1	PUF2	PUF3	Overall
Amorphous	47.88 %	51.38 %	48.77 %	49.34 %
Monocrystalline	51.49 %	50.52 %	46.39 %	49.47 %
Polycrystalline	50.97 %	52.08 %	50.74 %	51.26 %

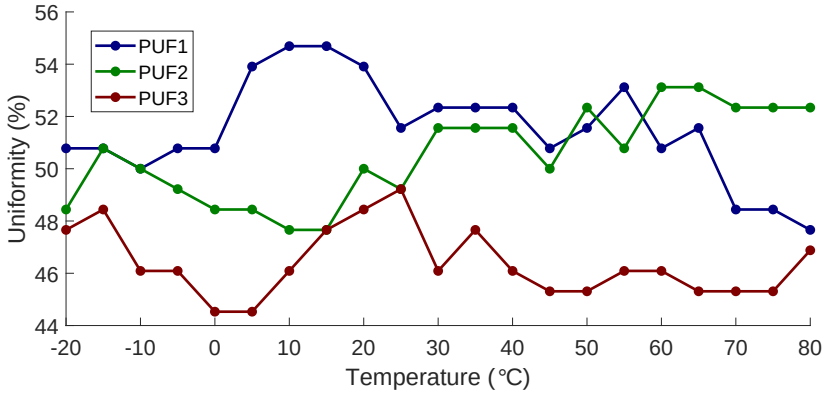


Figure 5.9: Monocrystalline silicon uniformity with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

5.5.2 Testing with Respect to Light Intensity

Unlike in standard CMOS ICs, light intensity plays a major role in controlling the electrical properties of solar cells. It is therefore very important to analyze the performance of our proposed PUFs against variations in light intensity. Testing was performed by using a variable transformer to vary the intensity of a LED bulb from $40Watts/m^2$ to $90Watts/m^2$. Readings were taken in increments of $5Watts/m^2$ and $40Watts/m^2$ was used as the reference for reliability testing. Our tested range was directly influenced by the testing facilities available to us. The complete testing setup is shown in Figure 5.11.

Reliability Testing

Figure 5.12 shows the reliability values of the amorphous silicon PUFs and Table 5.5 shows the average reliability values for each copy of the PUF across the measured range of light intensities. The three amorphous silicon PUFs produced a combined average reliability of 97.75% across the tested range.

Figure 5.13 shows the reliability values for each of the monocrystalline silicon PUFs and Table 5.5 contains the average reliability for each copy of the PUF. The first copy of the PUF (PUF1) demonstrated a sharper drop in reliability as light intensity increased when compared to the other instances of the PUF (PUF2 and PUF3). Overall, the monocrystalline silicon PUFs had a combined average reliability of 96.12%.

Figure 5.14 shows the reliability values for each of the polycrystalline silicon PUFs and Table 5.5 contains the average reliability for each copy of the PUF. The second copy of the PUF (PUF1) demonstrated a gradual decline in reliability as light intensity increased whereas the other instances of the PUF (PUF1 and PUF3) remained very consistent. Overall, the polycrystalline silicon PUFs had a combined average reliability of 95.45%

All of the PUFs tended to show resistance to changes in light intensity. The reliability tended to only gradually degrade as the light intensity increased with seven of the nine tested PUFs never dipping below 90%. The amorphous silicon has the highest overall

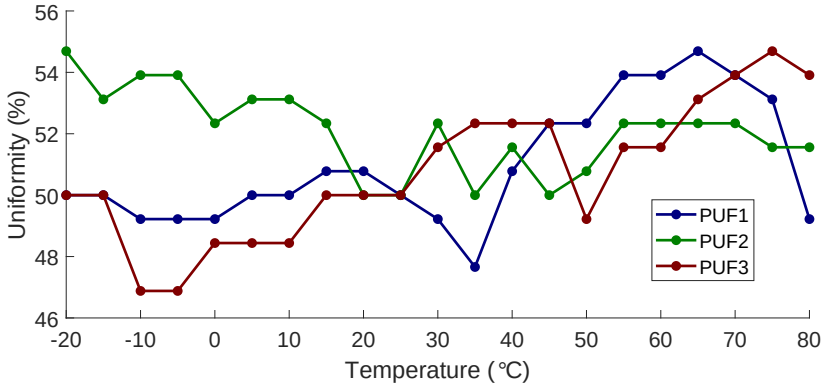


Figure 5.10: Polycrystalline silicon uniformity with respect to temperature. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

average reliability at 97.75% while even the worst one (polycrystalline) was still above 95% at 95.45%.

Table 5.5: Average Reliability with Respect to Light Intensity

	PUF1	PUF2	PUF3	Overall
Amorphous	98.37 %	96.31 %	98.58 %	97.75 %
Monocrystalline	93.32 %	97.44 %	97.59 %	96.12 %
Polycrystalline	97.30 %	93.39 %	95.67 %	95.45 %

Uniformity Testing

Figure 5.15 shows the uniformity values of the amorphous silicon PUFs and Table 5.6 shows the average uniformity values for each copy of the PUF across the measured range of light intensities. The three amorphous silicon PUFs produced a combined average uniformity of 50.00% across the tested range. There was little variation in the uniformity values for each PUF as light intensity increased. However, the second copy of the PUF (PUF2) was noticeably higher than the other two copies (PUF1 and PUF3).

Figure 5.16 shows the uniformity values for each of the monocrystalline silicon PUFs and Table 5.6 contains the average uniformity for each copy of the PUF. Overall, the monocrystalline silicon PUFs had a combined average uniformity of 52.81%.

Figure 5.17 shows the uniformity values for each of the polycrystalline silicon PUFs and Table 5.6 contains the average uniformity for each copy of the PUF. Overall, the polycrystalline silicon PUFs had a combined average uniformity of 48.74%

Variations in light intensity did not appear to have any consistent effect on the uniformity of the responses generated by each type of PUF. The uniformity value for each copy of the different PUFs remained close to the ideal value of 50% and the average uniformity

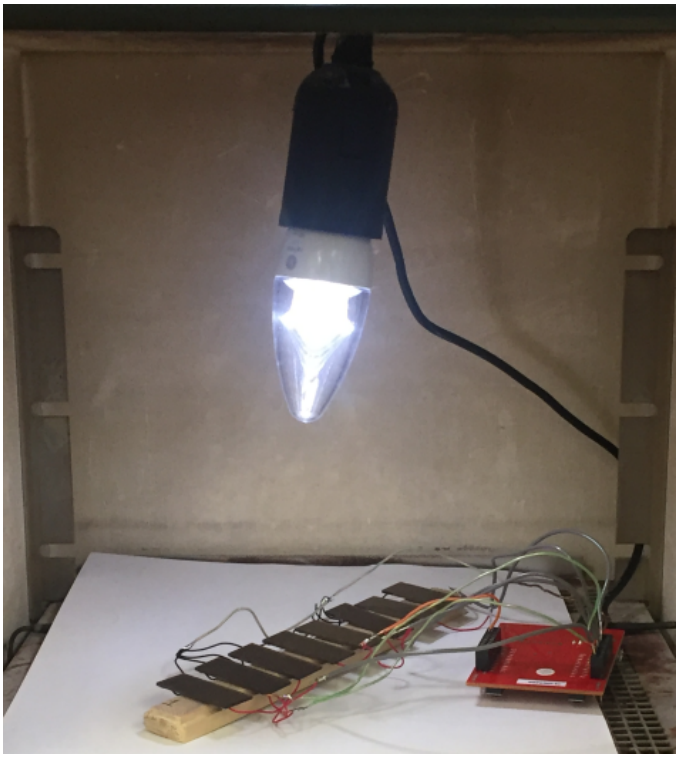


Figure 5.11: Light intensity testing chamber. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

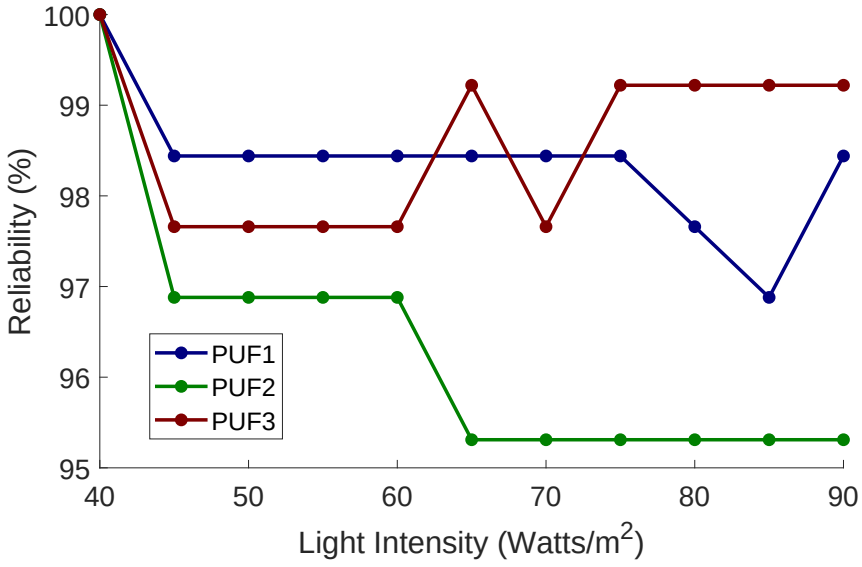


Figure 5.12: Amorphous silicon reliability with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

for the amorphous silicon PUFs was was the best exactly 50.00%. However, it is worth noting that this PUF also had the lowest recorded uniformity at 46.16% and the highest at 56.18%.

Table 5.6: Average Uniformity with Respect to Light Intensity

	PUF1	PUF2	PUF3	Overall
Amorphous	46.16 %	56.18 %	47.66 %	50.00 %
Monocrystalline	53.12 %	50.99 %	54.33 %	52.81 %
Polycrystalline	50.28 %	47.44 %	48.51 %	48.74 %

5.5.3 Uniqueness Testing

The uniqueness is a measure of how well a single PUF can be distinguished from the population as a whole. The ideal uniqueness value is 50% which effectively denotes that between any two PUF responses there is an equal probability that a given bit position between the responses will have unequal or equal values. It is calculated by using the following equation:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \tag{5.5}$$

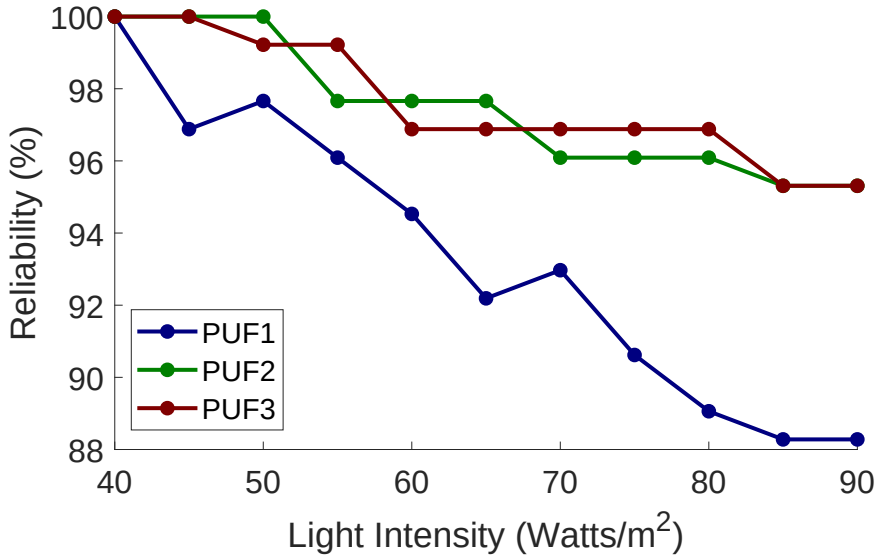


Figure 5.13: Monocrystalline silicon reliability with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

This equation effectively calculates the average hamming distance (HD) between every possible pair of PUFs in a population containing k total PUF copies. R_i and R_j are n-bit responses generated by PUF instances i and j, respectively such that $i \neq j$.

Uniqueness testing requires testing a relatively large number of PUF instances. Since creating that many physical PUF copies is not typically feasible, it is standard to perform Monte Carlo simulations in order to evaluate the uniqueness property of proposed PUFs. Unfortunately, the datasheets associated with our specific choices of solar cells only provide typical electrical parameter values. This makes it impossible for us to accurately generate the normal distribution of cells that would be required for a Monte Carlo simulation.

This closest approximation would be to use values for the the different solar cell materials that have been reported by other literature sources. Monocrystalline solar cells were previously reported to have a possible voltage range of $\pm 5\%$ [170]. The values reported in that work have been used by other works which have proposed methods for generating mathematical models of solar cell parameters [171]. Using this range we can generate a normal distribution of 8000 solar cells and randomly select 8 at a time to created 1000 simulated PUFs. The uniqueness calculated for the responses was 49.989% which is very close to the ideal value of 50%.

5.6 Discussion

In general, solar cells have been utilized as a power source for various IoT devices such as wireless sensors, RFIDs, etc. In this chapter, we have utilized the intrinsic variations between solar cells to create a PUF. Doing so effectively allows the solar cells to be used

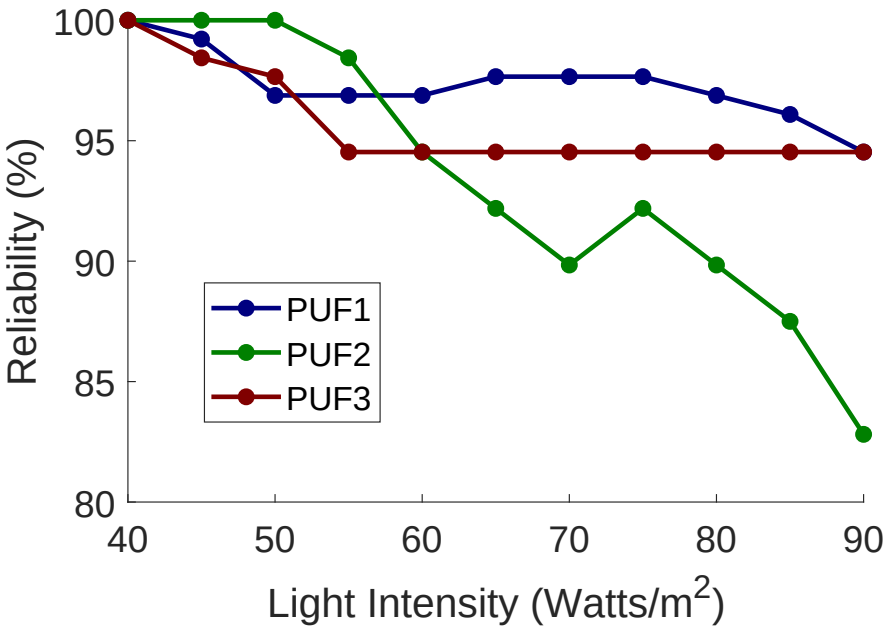


Figure 5.14: Polycrystalline silicon reliability with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

as a method for adding security features to these devices.

When compared to existing silicon PUFs, our design has the distinction that it could theoretically be implemented by a device without adding additional hardware. The integration of a silicon PUF would require a specially designed circuit which would not already exist on the device. Our proposed design methodology and Silicon PUF designs are not necessarily direct competitors despite the fact that they are both PUFs. Our proposed approach was specifically designed for applications where it was impossible to implement a Silicon PUF (e.g. adding security without adding hardware).

In this chapter we evaluated three different types of solar cell materials by performing temperature testing for the range -20°C to 80°C and light intensity testing for the range $40\ Watts/m^2$ to $90 Watts/m^2$. Each test was conducted on three copies of each type of PUF. We also evaluated the uniqueness of monocrystalline silicon through Monte Carlo simulations on a population of 1000 simulated PUFs.

In our experiments, we created PUFs using Panasonic AM-1417CA amorphous silicon solar cells ($V_{oc} = 2.4V$), IXYS KXOB22-12X1F monocrystalline solar cells ($V_{oc} = 630mV$), and AOSHIKE micro solar panel polycrystalline silicon solar cells ($V_{out} = 2V$). We created three copies of each PUF per type of solar cell for a total of nine PUFs. This allowed us to begin creating performance benchmarks for some popular types of solar cells when they are used in the creation of PUFs. Each copy was created from randomly chosen solar cells of each type. This means there is not anything purposely different between the different copies of a PUF per each type of solar cell. Any variation in testing results among

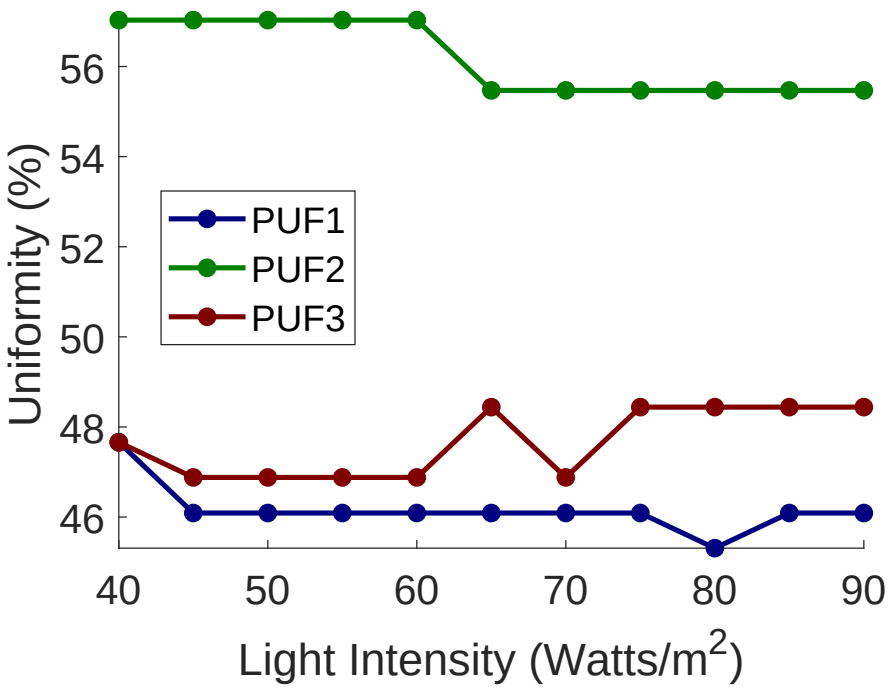


Figure 5.15: Amorphous silicon uniformity with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

the three copies of a given material should be a manifestation of their random intrinsic variations. This means that in tests where one PUF has noticeably different performance than the other two copies, those differences are the result of random chance rather than an explicit difference between the PUF copies. An example of this can be seen in the testing of reliability with respect to temperature for amorphous silicon where PUF1 had noticably different performance than PUF2 and PUF3.

On average, the polycrystalline silicon based PUFs had the best reliability with respect to temperature at 91.20% while the amorphous silicon based PUFs had the worst average reliability at 84.41%. Their standings are actually inverted with respect to Light Intensity as polycrystalline silicon had the worst average reliability at 95.45% and amorphous silicon had the best value at 97.74%. In addition, the uniformity values of the responses generated from these PUFs were also recorded and the average values for each type of solar cell were sufficiently close to the ideal value of 50%.

Based on the results of our testing, it can be inferred that polycrystalline solar cells are the ideal choice for PUFs that will be subjected to large variations in temperature. Our results also indicate that amorphous silicon solar cells are best suited for PUF applications where the major environmental concern is variation in light intensity. However, it is also worth considering the fact that the reliabilities with respect to light intensity were consistently higher than the reliabilities with respect to temperature. This would seem to imply that the PUFs have a higher resistance to light intensity variations than they do to temper-

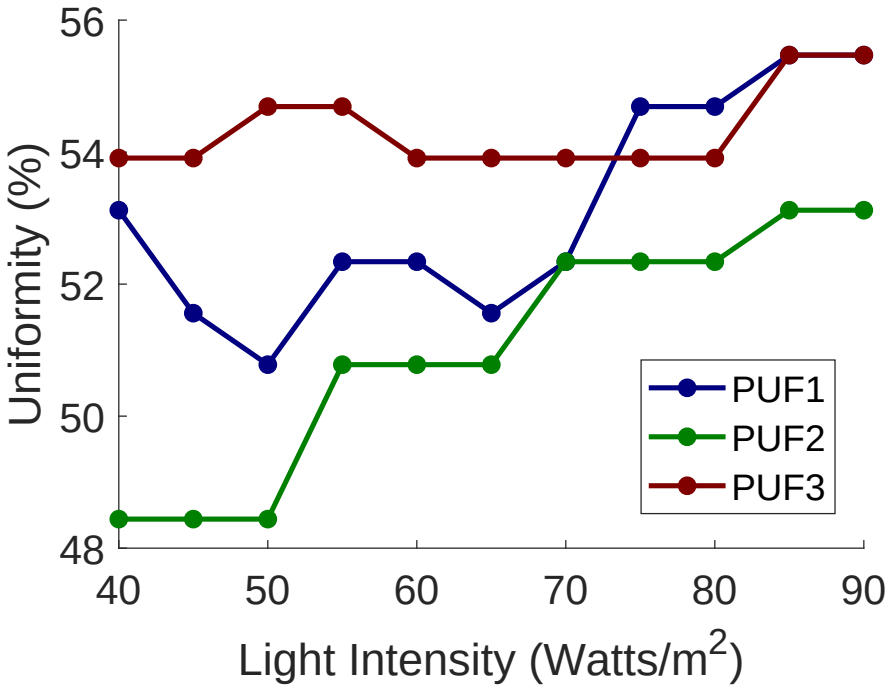


Figure 5.16: Monocrystalline silicon uniformity with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

ature variations for at least over the range that we tested. In an application scenario where variations in light intensity are the primary concern, special considerations must be made to ensure that any variations in temperature will be kept to an absolute minimum. Otherwise, seemingly minor changes in temperature could cause amorphous silicon to go from being the best option of our tested types of solar cells to the worst option.

Although we were not able to perform uniqueness testing on the specific cells used in our prototypes, we were at least able to evaluate the uniqueness of monocrystalline silicon solar cells. Through Monte Carlo simulation of 1000 simulated PUFs, it was determined that the uniqueness was 49.989%. While we can't draw any firm conclusions about PUFs constructed using amorphous silicon or polycrystalline silicon solar cells, the monocrystalline results are at the least not a discouraging sign for the uniqueness prospects of the other materials.

Based on the results of our testing we have noticed a curious trend related to the reliability of the cells with respect to temperature. The temperature coefficient of a solar cell is a measure how well the performs with respect to changes in temperature. The closer the coefficient is to 0% then the less the cell's output will drop as the temperature increases. Therefore, a lower temperature coefficient indicates better performance with respect to temperature. Among the materials we tested, researchers have shown amorphous silicon to perform the best as temperatures increase. That is followed by monocrystalline silicon solar cells with the next best performance, and polycrystalline had the worst performance [36]

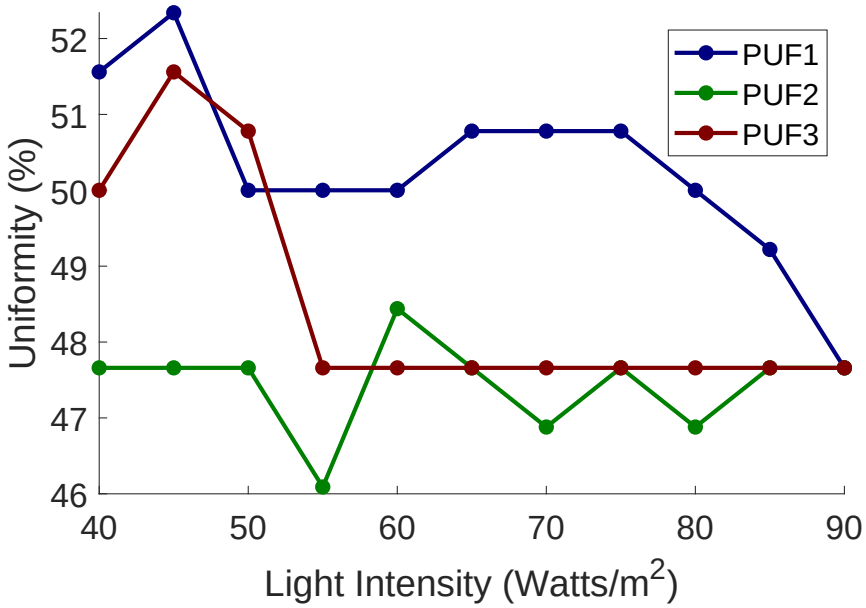


Figure 5.17: Polycrystalline silicon uniformity with respect to light intensity. Material from "C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems. SN Computer Science, published 2020, Copyright © 2020, Springer Nature Singapore Pte Ltd"

[38]. Based on these characteristics one would expect to see the PUFs constructed from these materials to exhibit the same relative performance when it comes to their reliability with respect to temperature. However, our testing has revealed the pollycrystalline PUF to have the best reliability with respect to temperature followed by monocrystalline and then amorphous. This is the exact inverse of their normal performance with respect to temperature. Determining what the reason is for the inverse correlation between temperature coefficients and PUF reliability could be further investigated as a future work.

5.7 Conclusion

In this chapter, we have proposed a design methodology to create PUFs from solar cells. This PUF uses the open-circuit voltages from the individual solar cells as the source of entropy in generating responses. Furthermore, we created copies of the PUF based around three different types of solar cells: amorphous silicon, monocrystalline silicon, and polycrystalline silicon. We evaluated the reliability and uniformity of our proposed design over a temperature range of -20°C to 80°C and a light intensity range of $40 Watts/m^2$ to $90 Watts/m^2$. From our testing we determined that the polycrystalline silicon PUFs had the highest average reliability with respect to temperature at 91.20% and the amorphous silicon PUFs had the highest average reliability with respect to light intensity at 97.74%. Furthermore, our Monte Carlo simulations on monocrystalline silicon showed that a PUF based on monocrystalline solar cells could have a uniqueness value of 49.989%.

The exploratory nature of this work presents multiple avenues for future work. There are other solar cell materials besides the three we evaluated which could similarly be suitable for the creation of PUFs. Our testing was performed with limited sample sizes. However, the results are encouraging and thus warrant further testing over greatly expanded sample populations. One potential future area of work would be performing more thorough investigations on a larger population. Large scale testing could be of interest for commercialization purposes where it would be useful to generate results that can be considered as accurate as possible for the entire population. Large scale testing would also allow for uniqueness testing to be performed on actual physical devices rather than simulated copies.

A further goal would be to improve the design to allow the solar cells to act as both a source of power and a source of entropy for the PUF. Standard solar cell usage only provides the former while our proposed design only allows for the latter. Combining the two should be feasible, but the actual circuit must still be designed. This would allow the PUF to effectively serve as a source of both power and security and ease its integration into IoT devices. As part of this integration, it would be worth exploring how best to integrate this behavior so that its operation does not run into conflict with the real time nature or other operating constraints of the devices.

Chapter 6

Fortifying Vehicular Security Through Low Overhead Physically Unclonable Functions

The adoption of PUFs would allow for the integration novel security solutions to cyber-physical systems. Consider a vehicle as one example of a cyber-physical system. The Controller Area Network (CAN) bus is a major method of communication between a vehicles critical systems. CAN was designed without any built in security features and adding traditional security features is not easily achieved without requiring major changes to the CAN protocol or the underlying hardware. Providing security thus requires developing completely new methods. In this chapter we propose a new security framework that adds security features while minimizing overhead and without making any changes to the basic CAN protocol. This framework is a server-based approach where a central server connected to the CAN bus is responsible for authenticating all connected nodes and generating session keys. The design utilizes physically unclonable functions (PUFs) as the basis for key storage, key generation, and the authentication of the nodes. Lightweight cryptographic algorithms are employed as they are more aptly suited than standard cryptographic algorithms to the resource constrained environments of vehicles. Figure 6.1 shows an example of the proposed framework incorporated into a smart vehicle environment.

The rest of this chapter is organized as follows: Section I introduces a major security concern in vehicles; Section II presents our vision for using PUFs as a low overhead smart car security solution; Section III describes related work on consumer electronics security and provides background information on CAN, its vulnerabilities, and PUFs; Section IV describes the proposed framework's operation in detail; Section V analyzes the framework and its security capabilities; Section VI provides a comparison to other PUF-based security frameworks; lastly, Section VII concludes the chapter.

6.1 Introduction

Vehicles are no longer a purely mechanical machine. Modern vehicles include a not insignificant number of digital components such as infotainment systems and the electronic

control units (ECUs) that are responsible for controlling the various subsystems within the vehicle. These devices are connected via various intra-vehicle networks, the most notable being the Controller Area Network (CAN) which provides a relatively inexpensive method for several ECUs to communicate with each other [39].

Unfortunately, CAN was not developed with security in mind. The lack of security has become much more alarming over the last decade as researchers have been able to successfully attack vehicles by exploiting the shortcomings in the CAN protocol. A very notable example of this was in 2015 when researchers were able to control a consumer vehicle [119]. Due to the nature of CAN, the ECUs for such systems as the engine, brakes, and steering were all connected to the same CAN bus. All an attacker needs to do to carry out an attack is gain access to the CAN bus. This could be achieved through somehow compromising an ECU or more simply creating their own connection. The lack of security features means that all transmitted messages are treated as being from a valid source regardless of their actual origin. For example, an attacker could send a message instructing the vehicle to apply the brakes. The vehicle would comply as it has no way of verifying the validity of the message.

The issue of vehicular security is likely to become even more pressing in the coming years. This in large part can be contributed to the continual push to develop fully autonomous vehicles in addition to the inclusion of smart features in vehicles. Every new type of connection added to a vehicle represents a new potential attack surface for malicious actors. Some of the connections include vehicle-to-vehicle (V2V), vehicle-to-network (V2N), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P). This overall connected environment is collectively referred to as vehicle-to-everything (V2X) [28].

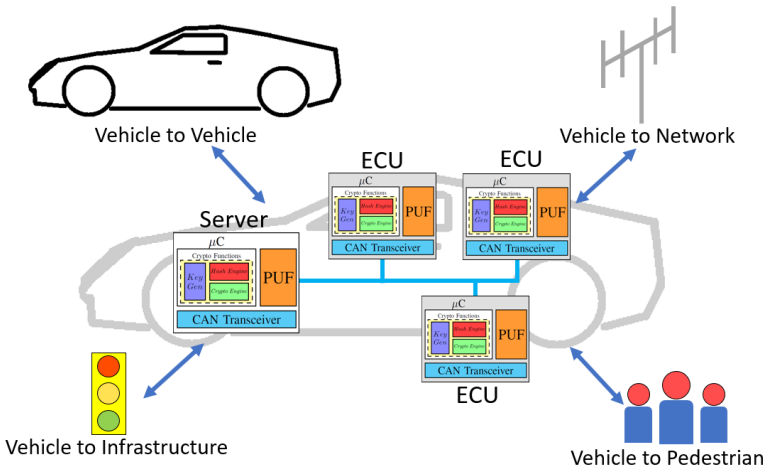


Figure 6.1: PUF Integrated Smart Vehicle

It is difficult to design a singular security solution since a vehicle's expanded features are provided by separate subsystems which communicate via in-vehicle communication networks. There in fact exists a knowledge gap in terms of how damage could by caused by attacking various components and systems of the vehicle. For example, it has been shown to be possible to compromise some of a vehicle's sensors in order to trick the driver and/or the vehicle's control system. However, it is unknown just how vulnerable all of the

sensors are and how severe of a reaction can be induced by a hacked sensor generating erroneous readings [132].

This is a constantly evolving problem that demands regularly devising new techniques to combat previously unknown vulnerabilities. However, vulnerabilities that have been known for years demand a similar level of time and focus. A major target should be devising solutions for the inherent vulnerabilities in the CAN bus. Because CAN is such a fundamental communication network, any potential security solutions should seek to remain as true as possible to the original specification. Major deviations could result in the need to redesign an untold number of internal systems to make them compatible with the new solution.

6.2 Our Vision for PUF-based Low Overhead Smart Car Security

We believe that the security challenges facing vehicles are so unique that classical security approaches alone will not be sufficient. Vehicles are designed such that some of their core functionality is directly provided by inherently insecure components and subsystems. These components are in fact so well ingrained that replacing all of them would likely require vehicles to be fundamentally redesigned from the ground up. While an approach like this could work in theory, the sheer cost of design, not to mention the material costs of the new components, would seem to prevent this from being a truly viable option for anything short of very long term goals.

In addition to providing security, we believe that for a security solution to be more immediately viable it should minimize both the monetary and computational costs that would be incurred by its introduction to the vehicle. As such, there are three major design goals that vehicle security solutions should strive to meet:

1. Minimize additional hardware and computation.
2. Avoid significant changes in protocols.
3. Minimize communication overhead.

First, security solutions should seek to minimize the addition of extra hardware and computation. The resource constrained and real-time nature of vehicles does not allow for much extra computation for things like encryption. Upgrading the existing devices or adding specialized hardware to provide the resources needed for the additional computation would drive up implementation costs.

Second, solutions should not make any significant changes to protocols. We consider significant changes to include any modification that would require likewise changes in the supporting hardware. For example, replacing a communication protocol or adding additional message fields cannot occur without upgrading the current infrastructure to support the new features.

Lastly, a solution should try to minimize communication overhead so that it does not effectively violate the second goal without actually changing the protocol itself. Some protocols like CAN can only send a very limited amount of data per message. Breaking a single transmission across multiple CAN messages allows for the transmission of cryptographic keys and encrypted data, but at the cost of reducing the bandwidth and responsiveness of the network.

Many potential security solutions would introduce additional overhead in both computation and the number of additional messages that must be sent in support of the normal transmission of data [182] [82] [147]. Furthermore, some solutions would likely require the addition of hardware for features such as secure key storage and generation, data encryption, etc. The use of physically unclonable functions (PUFs) in security solutions could potentially provide a cheaper option for the implementation of some of the these features. PUF-based security solutions would thus more closely align with our previously stated design goals. It is for that reason that our proposed security framework is directly based on PUFs. An example of this integration, which is used by our proposed framework, is shown in Figure 6.2. Every ECU would include its own PUF which could then be utilized by a variety of security operations. Integration of a PUF in this manner would leave open the possibility of maintaining the underlying CAN protocol and thus should not require any modifications to the actual CAN network infrastructure that connects the ECUs.

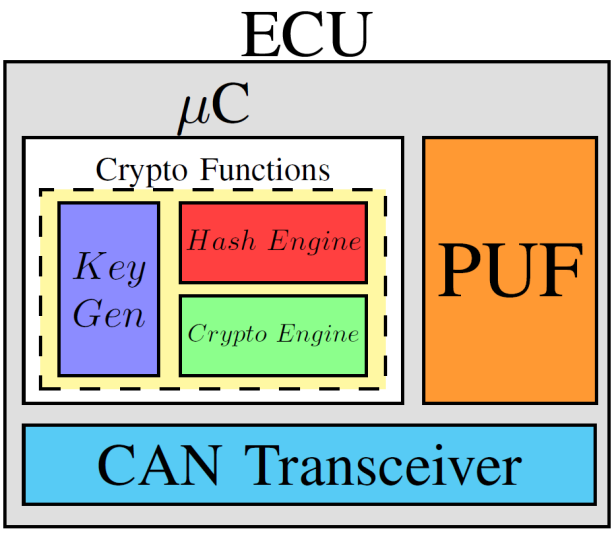


Figure 6.2: ECU PUF Integration

6.3 Background

6.3.1 Prior Related Work on Consumer Electronics Security

Providing security to vehicles is a challenging problem that is not readily solvable by conventional solutions. Even just the diagnosis of security threats has required the development of novel intrusion detection methods [121]. However, the security issues facing vehicles

are not as isolated as they might appear. Similar security concerns are actually being raised in a variety of other areas. This general trend is a direct response to society's adoption of the Internet of Things (IoT). The addition of smart features to an increasing number of consumer electronics has also introduced security vulnerabilities and concerns that were not present when the devices were originally designed.

Developing methods to combat these new challenges has drawn interest from a number of researchers. This has included classical approaches such as designing hardware security chips for mobile devices [74] and secure firmware validation and update schemes for personal home devices [31]. Other areas of interest include security architecture for edge devices [163] and protecting the runtime data of embedded systems through hardware-enhanced cryptographic engines including AES and the hashing algorithm LHash [174].

Researchers have also taken to examing more novel security approaches such as creating PUFs that are specifically designed for use in IoT applications. This has included both adaptations of established designs such as Ring Oscillator (RO) PUF [80] along with novel approaches such as designs based on adiabatic logic [90] and bloom filters on memristor-based PUFs [100]. Researchers have explored how PUFs such as these could serve as the basis for more complete security frameworks and systems. One interesting example is a framework in which individual embedded devices use PUFs to create their own unique fingerprints [62]. Those fingerprints are then encoded in order construct a larger system-level fingerprint. In this way the system level ID can be used to identify if one of the system's individual devices is no longer valid. Another approach utilizes memristor-based PUFs to create a very lightweight security system [167]. The PUFs operate as a one time pad by generating a random key each time one is needed for an encryption and decryption operation. A random response is sent to the PUF and the associated response is used as the key. Other research efforts have included using PUFs to create novel device authentication schemes for IoT-enabled medical devices [187] and radio-frequency (RF) communication between nodes in a wireless network [24]. The inclusion of PUFs has the potential to thus introduce security features into an intra-vehicle network while minimizing any changes in its normal operation.

6.3.2 Controller Area Network (CAN)

The Controller Area Network (CAN) is a serial communication system that allows for simple and efficient message passing between connected nodes without requiring a master controller in the network [39]. CAN is most commonly used in vehicles to allow communication between the embedded electronic control units (ECUs) without having to implement point to point wiring between all possible communication paths. Figure 6.3 shows the format of a standard CAN frame or message. A standard CAN frame has a very limited number of message fields. The arbitration portion denotes the ID of the message. The control field shows the number of bytes of data (0-8 bytes) being sent by the frame. CRC stands for cyclic redundancy check and is an error correcting code used to check for errors in the transmission. ACK is used to denote if a message was successfully received. Lastly, EOF denotes the end of the frame.

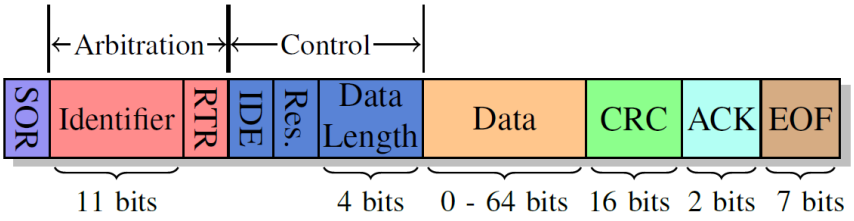


Figure 6.3: CAN Frame

6.3.3 CAN Vulnerabilities

The CAN protocol was not originally designed to include much in the way of security features. The key issues are messages are broadcast to all connected nodes, the data fields are not encrypted, and there is no way to authenticate or even known who was responsible for sending a given message. An attacker only has to gain access to the CAN bus in order to carry out a wide arrange of attacks including eavesdropping, spoofing/impersonation, and denial of service (DoS). Through eavesdropping an attacker would be able to monitor all communications and launch a replay attack by sending a duplicate of a previously seen message [86]. Another possibility would be to reverse engineer what would be the (likely manufacturer-specific) communication protocol used between nodes. Once that was accomplished, an attacker would be able to send erroneous messages that the targeted ECUs would interpret to be valid due to CAN's inherent lack of authenticity. Researchers have shown that attacks of this nature can be utilized to control different components of the vehicle such as controlling the dashboard and shutting off the engine [181].

The CAN protocol also makes CAN very susceptible to DoS attacks. The CAN standard guarantees that the message with the highest priority will be the first message to go through. If the CAN bus is currently in the process of transmitting a message, it will stop that transmission and begin to transmit the new message provided that new message has a higher priority. An attacker only has to repeatedly transmit high priority messages for the CAN protocol to guarantee that the messages from ECUs will never get a chance to send due to having a lower priority [23].

6.3.4 Physically Unclonable Functions

Physically Unclonable Functions (PUFs) are a class of device that utilize internal variations introduced by the manufacturing process to generate unique outputs for a given input. The input to a PUF is denoted as a "challenge" and the output is known as a "response". A challenge and its associated response are collectively known as a challenge-response pair (CRP). For a given challenge, the response produced by different PUFs should be unique since each response is a direct manifestation of the unique physical properties of that specific PUF. Furthermore, a PUF with a small number of CRPs, typically just one, is a weak PUF and a PUF with a large number of CRPs is considered to be a strong PUF.

PUF designs are commonly based on transistor level process variations such as gate delays [61] or the initial power-on value in memory cells [118]. Other researchers have explored creating PUFs from larger components such as energy harvesters [129] and sensors

[98]. The unique properties of PUFs make them an intriguing option as a low cost method for implementing security related features such as key storage [45] or hardware obfuscation [176].

6.4 Proposed CAN Security Framework

The overall design of our proposed framework involves using a server within the network to authenticate all nodes before allowing normal message passing operations to begin. The proposed framework requires an LWC functions for encryption, decryption, and hashing. Any LWC function can be used as long as it meets certain criteria. The LWC function used for encryption and decryption must have a block size of 64 bits and a key size of no more than 128-bits. For the LWC hash function, it must be able to generate 128-bit hashes.

As an example, our proposed framework is described in terms of using PRESENT [21] for encryption and PHOTON [56] for hashing. We use these as examples as they have both been defined as International Organization for Standardization (ISO) standards for LWC[68] [69]. Either LWC function could however be substituted with a different one which meets the aforementioned criteria. The proposed framework also makes use of Elliptic Curve Diffie-Hellman (ECDH) key exchange based on FourQ which has been shown to offer better performance than other curves targeting the same level of security [35]. These cryptographic algorithms will be discussed in more detail in Section 6.5. The proposed framework supports 80-bit or 128-bit encryption keys. For the sake of simplicity, the figures and tables in this section assume 80-bit encryption keys.

Our proposed framework is not designed for use with only one specific PUF design. It is assumed that the chosen PUF will be a weak PUF since the framework needs a given PUF to always produce the same response each session. The keys are derived from the PUF responses so the keys would change if the response changed. A strong PUF could be an option if it was configured to operate as a weak PUF by always providing it the same response. Topics related to the actual implementation of the PUF should be considered outside the scope of this chapter. This includes methods for improving the reliabilities of PUFs such as error correcting codes and other schemes. Additional resources required for a specific PUF implementation are likewise a direct result of the chosen PUF rather than our proposed framework.

The proposed framework can be divided into the distinct operation phases of enrollment, authentication, and normal operation. The authentication and normal operation phases will occur every time the system is turned on. By contrast, enrollment would ideally only ever occur once for the entire existence of the system. The rest of this section describes each of the phases in greater detail.

6.4.1 Enrollment

This phase should only occur once, likely during the manufacturing phase. This should in theory provide a secure environment for data to be hardcoded into the server and other nodes. The purpose of the enrollment phase is to give each node a copy of the server's public key. This allows each node to ultimately derive a shared secret with the server that

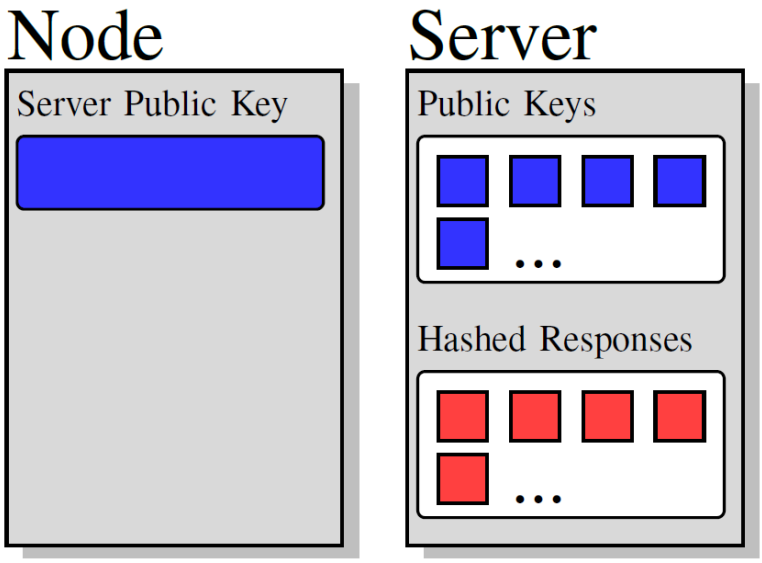


Figure 6.4: Post-Enrollment Stored Values

it can use to securely communicate with the server during the authentication phase. The server will likewise need to have a copy of the public key for every node. In addition, the server needs to store a hash of the response from each node. The response hashes are 128 bits in size which means future stages will only need 2 CAN frames to transmit the entire hash.

We use response hashes rather than raw responses for two main reason. The first reason is it allows greater flexibility in choosing a type of PUF to use within the framework. Choosing a PUF with a response larger than 128 bits won't increase the number of CAN frames required for a node to send it to the server. The other reason is this prevents sending the PUF's response outside of the node. Even though the responses would be encrypted, the server would still need to do a comparison with the unencrypted response in order to validate it. The raw response can be used to directly generate a secret key, while it is not possible to do the same with a hash of the response. This removes the need to take the same security precautions with storing and handling the response hash that you would need if you were instead using the raw response.

During authentication a node will be considered valid if it is able to generate a response whose hash matches the associated one stored by the server. Figure 6.4 provides a visualization of what data will be stored within each entity at the end of the enrollment phase.

6.4.2 Authentication

The authentication phase should run every time the network is first powered on. Within this phase the server will first validate the authenticity of each node in the system. Next it will generate a session key and send a hashed copy to each node to use during normal operation. Algorithm 3 describes the individual steps taken by a given node and Algorithm 4 describes the steps for the server. These steps assume 80-bit encryption keys. In addition, the entire authentication process is illustrated in Figure 6.5.

Algorithm 3 Node Authentication Process

1: The node's PUF generates a response $\mathbf{R}$.
2: A 128-bit hash of the response is created $\mathbf{H_R}$. The response is also used as the node's secret key $\mathbf{x}$ by FourQ.
3: A shared secret $\mathbf{SSec}$ between the node and the server is generated using the node's private key $\mathbf{x}$ and the stored public key of the server $\mathbf{P_S}$.
4: The shared secret is hashed and truncated to produce an 80-bit key $\mathbf{K_{SSec}}$.
5: The node's hashed response $\mathbf{H_R}$ is encrypted using the hashed shared secret as the key $\mathbf{K_{SSec}}$.
6: The encrypted response hash is sent to the server.
7: The node then waits for the server to respond with an encrypted session key.
8: The node decrypts the session key $\mathbf{K_{Sess}}$ using the hashed shared secret as the key $\mathbf{K_{SSec}}$.
9: The list of valid nodes is extracted from the decrypted session key if the system was configured to support it.
10: This session key $\mathbf{K_{Sess}}$ will later be used during normal operation to encrypt and decrypt all messages within the network.

Algorithm 4 Server Authentication Process

1: The server's PUF generates a response $\mathbf{R_S}$.
2: The response is used as the server's secret key $\mathbf{x_S}$ by FourQ.
3: A shared secret $\mathbf{SSec}$ between a given node and the server is generated using the server's private key $\mathbf{x_S}$ and the stored public key of the node $\mathbf{P}$.
4: The shared secret $\mathbf{SSec}$ is hashed and truncated to produce an 80-bit key $\mathbf{K_{SSec}}$.
5: The server waits to receive encrypted response hashes from each node.
6: The server decrypts the hashes using the hashed shared secret associated with that specific node as the key $\mathbf{K_{SSec}}$.
7: The decrypted response hashes $\mathbf{H_R}$ are validated by comparing them to previously stored hashes.
8: The server generates a random session key $\mathbf{K_{Sess}}$ and concatenates it with either padding or a bit mask representing valid nodes in the network.
9: The server encrypts a copy of the concatenated session key $\mathbf{K_{Sess}}$ for each node using the hashed shared secret associated with that node as the key $\mathbf{K_{SSec}}$.
10: The server sends an encrypted session key to each node.

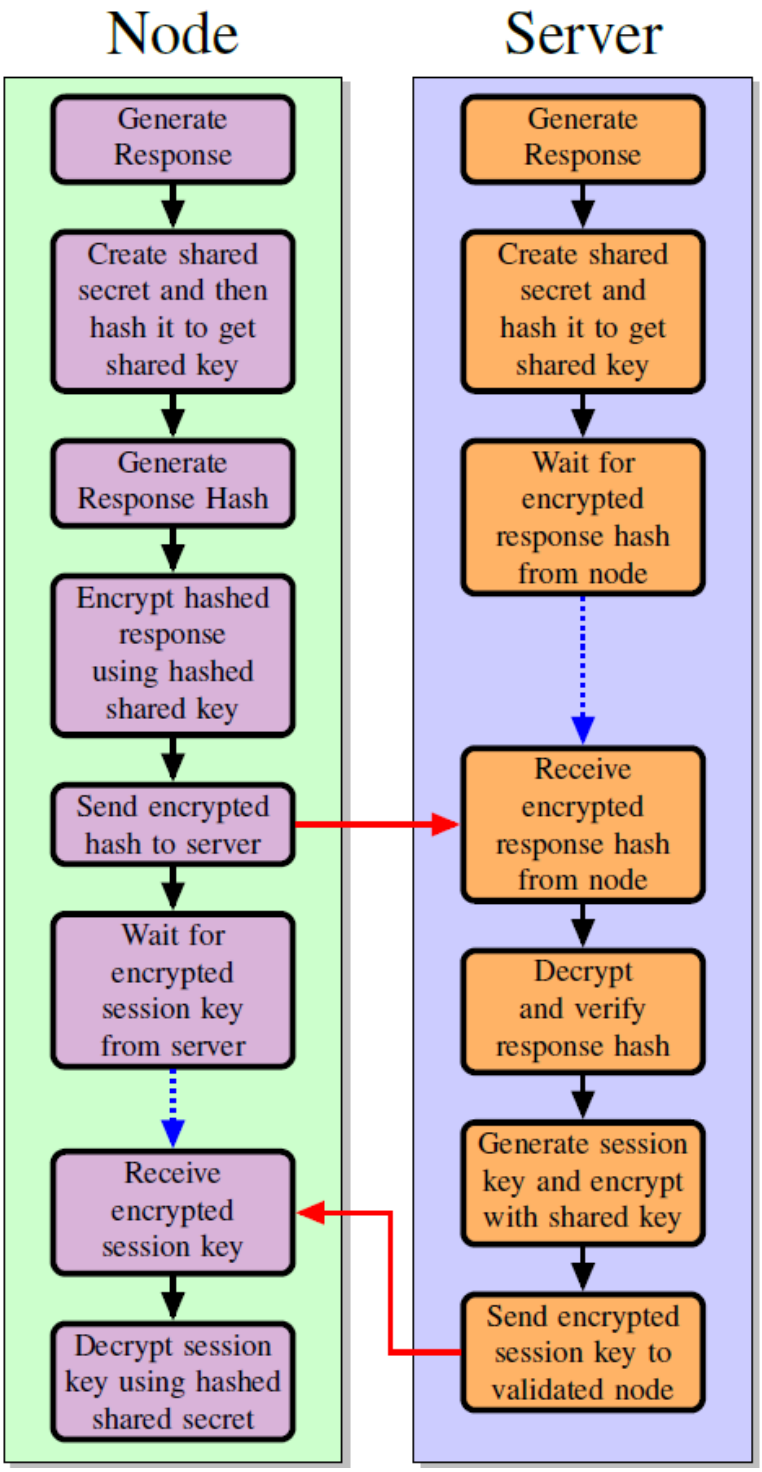
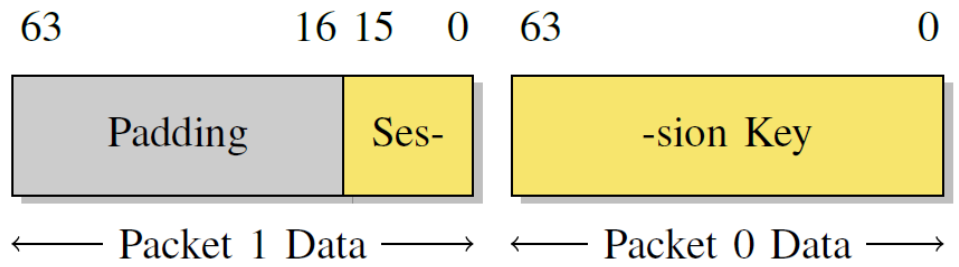
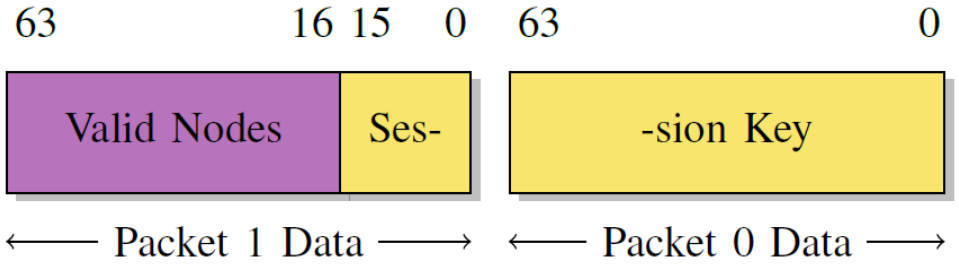


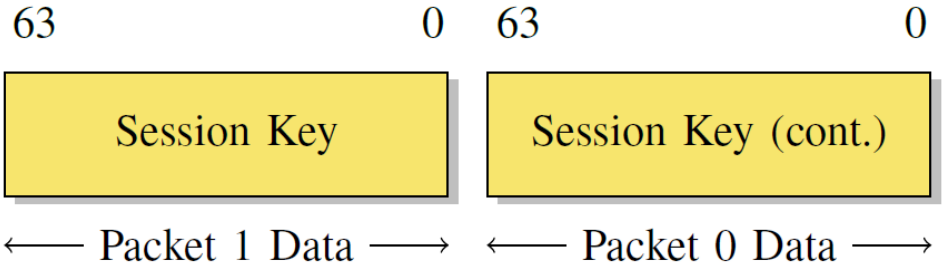
Figure 6.5: Authentication Process



(**a**) 80-bit Session Key



(**b**) 48-bit Valid Bitmask and 80-bit Session Key



(**c**) 128-bit Session Key

Figure 6.6: Options for Encrypted Session Key Packets

It is important to note that the shared secret between each node and the server is not directly used as a key for encryption and decryption. We hash the shared secret to get a shared key. This helps prevent key leakage and reduces the shared secret to the key size required by the encryption algorithm. The nodes use the shared key to transmit their response hashes and the server uses it to transmit the session keys.

The only messages sent during this phase are the encrypted response hashes and the encrypted session keys. The encrypted response hashes are 128 bits in size which means it will take 2 CAN frames to transmit the entire hash. Similarly, the encrypted session key must be a multiple of 64 bits in order to minimize the amount of CAN frames required to transmit it. This allows us to support key sizes of 80 and 128 bits. 80-bit keys would need to be concatenated with 48 bits of padding. Alternatively, an 80-bit key could be concatenated with a 48-bit wide bitmask that denotes the nodes that were successfully Authenticated. Each bit would correspond to a specific node. Figure 6.6 provides an illustration of these different modes of operation.

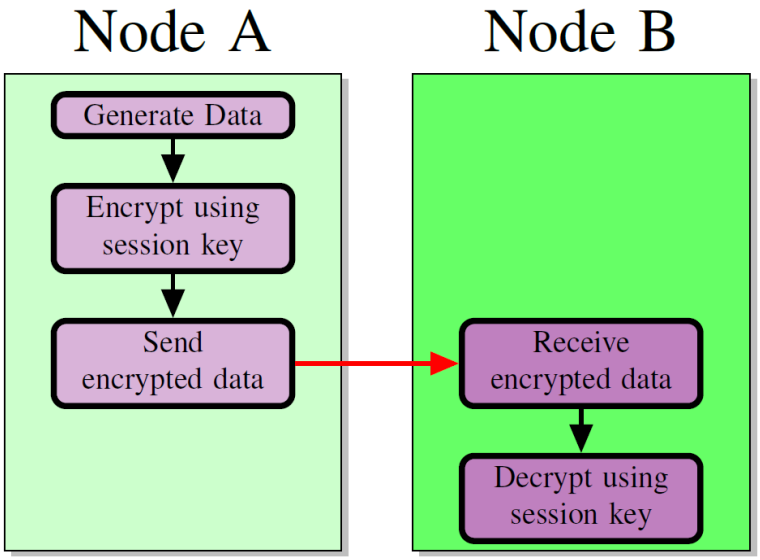


Figure 6.7: Normal Communication Between Two Nodes

6.4.3 Normal Operation

This phase is analogous to the way a normal CAN bus operates and ultimately serves the same purpose. The major difference is all transmitted data must be encrypted before it is sent over the bus. Once nodes have have been authenticated by the server and received a session key, the system can transition to normal communication between nodes in the network. The session key is used to encrypt the data field of a packet before sending it across the CAN bus to another node. That other node can then use its own copy of the session key to decrypt the data and then respond accordingly. Figure 6.7 shows the general flow for one node communicating with another node.

6.5 Design and Analysis of Proposed Framework

By deriving keys from physically unclonable functions (PUFs), we avoid the need to use costly secure nonvolatile memory for key storage. Instead, the keys can be generated as needed during each authentication phase. This means an attacker would have to obtain physical access to the PUF to recover its response and associated key pair. As we will explain, the information that must be persistently stored between sessions does not necessarily need to be kept secret and that allows us to save costs by not requiring secure nonvolatile memory in the nodes.

The public keys can be stored in unsecured memory since they require a separate private key to form a shared secret. The private key is generated whenever it is needed and deriving a private key from its associated public key would require successfully breaking the elliptic curve key generation cryptographic algorithm. The hashed responses can also be stored in unsecured memory since the server expects any received hashed responses to be encrypted with the appropriate shared secret that it is assumed an attacker is not able to obtain. Furthermore, since hash functions are considered to be one-way it should not

be possible to recover the original input response that produced the hash and then use that response to generate its associated private and public keys.

During each new session, the server generates a session key that will be used by all ECUs during normal communication. Depending on the mode of operation, that key can be concatenated with a bit mask denoting which nodes are valid. During authentication a server with n ECUs will receive n hashed responses and then only have to transmit n total copies of the generated session key, one for each ECU. Therefore, the number of frames that must be sent in our proposed framework scales linearly as the number of ECUs in the system increases. Furthermore, the hashed responses and session key payloads can each be transmitted in only 2 frames. This means that during the authentication phase of our proposed framework, a network with n ECUs will require the transmission of $4n$ total frames to complete the authentication phase.

The overhead in terms of frames required by the proposed framework is shown in Table 6.1. The enrollment phase is omitted since it would likely not require sending any messages over the CAN bus and would ideally only ever run once. A partial repeat of the enrollment phase would only need to occur when PUFs are added and/or removed from the system, e.g. completely replacing a malfunctioning node. The enrollment phase is otherwise completely implementation dependent and occurs outside of the flow of operations for the system. During normal operation, our proposed framework operates exactly the same as the standard CAN protocol. The same number of messages are required to transmit the same amount of data. The only real difference is the data contained within the data field of the message is now encrypted. The cryptographic operations will of course introduce some additional overhead, but that will be highly dependent upon the chosen algorithms and the underlying hardware. Special purpose hardware for example could greatly speed up calculations or certain cryptosystems may perform better on the specific ECUs used by a given manufacturer.

6.5.1 Threat Mitigation

The major threat that will be directly mitigated is eavesdropping. Currently an attacker with access to the CAN bus can see all messages that are transmitted. Our proposed framework counteracts this by encrypting the actual data that is transmitted. The only potentially useful information that could then be used by an attacker is the destination IDs of the messages.

Other notable attacks are data tampering and impersonation attacks. As the name suggests, data tampering attacks occur when an attacker is able to successfully modify a message without the sender or receiver being able to detect that the message has been changed. Impersonation attacks are where an attacker impersonates another ECU and sends messages as if they were that ECU. These attacks are especially concerning because they can allow an attacker to effectively control a vehicle. The CAN protocol has no built in mechanism for identifying the original sender of any message. An attacker for example could send messages to engage the brakes and the brakes would activate as if the associated ECUs had received legitimate commands. Our proposed framework also provides some protection against these sorts of attacks. The first step of being able to forge messages is to understand the actual message format. Doing so requires an attacker to reverse engineer the message protocol by monitoring the network. If the actual message format is not already

known by an attacker, then it will be difficult for them to reverse engineer it since the data itself will be encrypted within our framework.

In the event an attacker does know the message format, it will still be difficult for them to create erroneous messages to produce specific outcomes like applying the brakes. All messages are encrypted with a session key so an attacker would need to have a copy of that key in order to properly encrypt their message. Otherwise, their message will get mangled when the receiving node attempts to decrypt it. This should force the attacker to resort to a replay attack in which they capture a message and then repeat it to produce a known result. This is much more time consuming since the attacker would have to try to monitor the entire network traffic and somehow correlate a specific message payload to a specific ID with producing a desired response in the vehicle. Since the session key is randomly generated each time, the encrypted form of a given message will change each time the session key changes. The attacker would then have to repeat the entire process every time a new session begins. This prevents an attacker from simply building a library of messages across several sessions since the messages will change each time.

The types of security threats that our framework does not offer much protection against are those that do not require reading and/or writing specific data values. Attacks of this nature succeed by merely transmitting a message regardless of its actual content. One notable example of this type of attack is a Denial of Service (DoS) attack. A DoS attack would seek to disable a vehicle by flooding the CAN bus with high priority messages. The higher priority means that these erroneous messages will get delivered before the valid, yet lower priority messages required for normal operation. The valid messages never get delivered and the vehicle is thus unable to function. Certain forms of data tampering and impersonation attacks would also fall under this type of attack. The goal of these attacks would not be producing a specific outcome such as controlling the vehicle's movement. Instead, they would seek to cause general havoc by either repeating previously seen packets or sending what would amount to junk data to ECUs. The attacker would have no notion of what the outcome will be. It would instead be completely up to chance in terms of how the vehicle will actually respond. As such, the possible response could range in severity from effectively ignoring the attacker's messages, all of the way to actually causing some sort of accident.

6.5.2 Cryptographic Algorithms

The performance of current cryptographic standards is not always suitable for use in resource constrained environments. Lightweight cryptography (LWC) seeks to address this by specifically designing cryptographic algorithms for resource constrained environments [162]. Although the National Institute of Standards and Technology (NIST) is currently in the process of setting LWC standards, the International Organization for Standardization (ISO) has published LWC standards.

Our adherence to lightweight cryptographic algorithms [68] [69] should provide other performance benefits and potentially reduce the cost of implementation. The amount of computation required to perform basic cryptographic operations such as encryption, key generation, etc., is reduced in our proposed framework compared to existing solutions which utilize larger algorithms such as AES. This has the added benefit of potentially

simplifying any dedicated hardware that is solely designed to perform those operations. Furthermore, our design does not require any form of secure nonvolatile memory for key storage as the use of a PUF allows all keys to be generated as needed.

ECDH based on FourQ

For key exchange we used Elliptic Curve Diffie-Hellman (ECDH) based on FourQ. FourQ is an elliptic curve which targets the 128-bit security level [35]. Although other curves targeting the 128-bit security level would also work, FourQ has been shown to be faster than other popular 128-bit security elliptic curves such as NIST P-256 and Curve25519 in both key generation and secret exchange [35] [4].

Encryption and Decryption

The use of a block cipher which has a block size of 64 bits means that during regular communication, the number of CAN frames that must be used to send encrypted data will remain the same as the number that must be used to send normal unencrypted data. Any additional overhead introduced by the proposed framework during normal operation would thus be solely limited to the encryption and decryption operations performed by each node.

Hash Function

It was important to choose a hash function that produced 128-bit hashes as it would allow us to use hashes as encryption keys. The other major benefit is hashed responses can be sent in just 2 CAN frames.

6.5.3 Server Capabilities

It is assumed that the server will be secure and have the capability to securely generate a random session key for each session. It should not be possible to add, remove, and/or modify nodes and the public keys and hashed responses associated with them except during the enrollment phase in a trusted environment. The server also has the potential to act as a monitor of sorts during the authentication phase. It could phase certain anomalies as malicious and either lock out the rest of the authentication phase, or notify a more centralized security system so that it may act accordingly. The CAN protocol does not show the origin of messages being transmitted. However, it can still detect situations such as multiple authentication attempts for a single node, an incorrect number of nodes attempting to authenticate, or false messages being transmitted before authentication has ended.

6.6 Comparison to Existing Designs

The security holes present in the CAN protocol have led to led researchers to propose a variety of different possible solutions. These approaches tend to involve adding security features through either changes to the base CAN protocol itself [51] [135] [1] or proposing frameworks around the existing protocol (such the one we are proposing) so that the CAN

protocol itself remains the same [182] [82] [147]. To the best of our knowledge, there are not many proposed security solutions that explicitly integrate PUFs as a core component of the system. As such we are not considering systems in which a PUF could replace an existing component such as using a PUF to remove the need for secure nonvolatile memory [45].

One example PUF-based solution is the work from [1]. That work uses PUFs embedded in each ECU to validate the ECU before sending a message to another ECU. All communication between ECUs is routed through a reference monitor which is responsible for validating the identity of the ECU during each communication before forwarding the associated message to its intended recipient. We are not considering this work in our comparisons due to the fact that the CAN bus has been effectively replaced by the reference monitor.

A separate work also uses a PUF and server based approach in which the server and each ECU have an integrated PUF [147]. During authentication, the server and each ECU generate ECDH key pairs from responses generated by each PUF. Every ECU generates a ECDH key exchange shared secret with the server and transmits an encrypted copy of its public key. AES-128 is used for encryption the keys. The public key is transmitted across two CAN frames since AES requires block sizes of 128 bits and the data field in a single CAN frame is only 64 bits. The server compares the received public keys with the public keys that it stored during an enrollment phase to validate each ECU. The server then sends encrypted copies of each valid public key (2 frames each) to each valid ECU along with a third frame denoting which ECU is associated with that public key. The ECUs use the received public keys to generate a shared secrets with every other valid node. The ECUs can then encrypt data being sent to any ECU with a unique key that is only available to the sending and receiving ECUs. Like before, two CAN frames must be sent for every data transmission to comply with the block size of AES-128.

There are certain scalability, functionality, and security concerns present in the existing framework that our proposed solution is able to overcome. The scalability issue lies with the authentication phase. Each valid ECU must receive encrypted copies of the public keys for all other valid nodes. For a system with n ECUs, a single ECU will transmit its public key to the server and receive an encrypted copy of each valid public key along with a message indicating the node ID associated with each key. This means the server must overall transmit n^2 public keys and therefore the number of public keys that must be sent will scale quadratically as the number of ECUs increases. If you consider that each public key sent by an ECU requires 2 CAN frames and each public key transmitted by the server requires 3 frames, then the total number of frames required for authentication is $3n^2 + 2n$. In addition, the fact that there is a unique shared secret between every pair of ECUs prevents broadcast messages. An ECU must separately encrypt and send duplicate messages to each intended ECU.

As stated in the previous section, our proposed framework will complete authentication after sending $4n$ frames for a system containing n ECUs. This means the number of required frames scales linearly with the number of ECUs in the system. The fact that there is a single session key shared by all of the nodes mean that our proposed framework supports messages having multiple intended recipients. The use of PRESENT for encryption allows an entire encrypted message to fit within the data field of a single CAN frame during nor-

Table 6.1: Required CAN Frames for n ECU System

Operation Phase	**[147]**	**Proposed**
Authentication	$3n^2 + 2n$	$4n$
Normal Communication	2	1

mal communication between ECUs. Table 6.1 shows a comparison between [147] and our proposed framework in terms of the total number of CAN frames that must be sent during the different phases of operation within a system containing n ECUs. The table shows that our approach scales much better for larger systems. For example, a system with 20 ECUs would require the transmission of 1240 frames under the existing framework while our proposed framework would only require 80.

This scaling issue becomes even more important when you consider the amount of time it actually takes to transmit a CAN frame. CAN has both high-speed and low-speed versions. High-speed CAN can transmit data at speeds of up to 1 Mb/s while low-speed can transmit at speeds of up to 125 kb/s. Standard CAN frames are 108 bits long. There is also an extended version which is 128 bits. Furthermore, CAN requires at least 3 bits of spacing between messages. This effectively means that standard and extended frames require the transmission of 111 bits and 131 bits, respectively. Therefore, Low-Speed CAN can transmit a standard and extended frames in 896 μs and 1048 μs, respectively. High-Speed CAN can transmit the frames in 112 μs and 131 μs, respectively.

Figures 6.8 and 6.9 show how long the Authentication would take as the number of nodes increases. Figure 6.8 assumes the system is using standard CAN frames and Figure 6.9 assumes extended CAN frames. For a system with 20 ECUs, our proposed framework will complete authentication in only 6.5% of the time that it would take the existing framework. That percentage will continue to decrease as the number of ECUs increases. Table 6.2 contains a comparison of the time required to transmit all of the CAN frames required to complete Authentication with in systems of various sizes. It is important to highlight the amount of time required for Authentication since it represents extra overhead that is not already present within vehicles. Implementing these frameworks would require introducing a period of time that the vehicle is unresponsive immediately after it starts. This period of time might be negligible for systems with a very small number of ECUs, but it will become increasingly pronounced as the number of ECUs increases. Most importantly, the superior scaling of our proposed framework guarantees that this dead period of operation will remain nearly imperceptible for much larger systems compared to the existing approach.

NIST guidelines state that for symmetric keys of size 128 bits, the elliptic curve key size to provide equivalent security is 256 bits [14]. Per the NIST specifications, security for the 128-bit ECDH key used in [147] would actually be equivalent to a symmetric key that is less than 80 bits. The normal security strength of AES-128 is potentially undercut since the shared secret used as the encryption key is derived from the 128-bit ECDH keys. This could present a vulnerability that could be exploited by an attacker. In our approach, the encryption key used during normal operation is a session key that the server randomly generates each time. During enrollment, we are able to make use of 256-bit ECDH FourQ

Table 6.2: Time Required to Transmit the Frames Required for Authentiction

Speed, Frame	Framework	Number of ECUs				
		5	10	15	20	25
High, Standard	[147]	9.52 ms	35.84 ms	78.96 ms	138.88 ms	215.6 ms
	Proposed	2.24 ms	4.48 ms	6.72 ms	8.96 ms	11.2 ms
High, Extended	[147]	11.134 ms	41.92 ms	92.36 ms	162.44 ms	252.18 ms
	Proposed	2.62 ms	5.24 ms	7.86 ms	10.48 ms	13.1 ms
Low, Standard	[147]	76.16 ms	286.72 ms	631.68 ms	1,111.04 ms	1,724.8 ms
	Proposed	17.92 ms	35.84 ms	53.76 ms	71.68 ms	89.6 ms
Low, Extended	[147]	89.08 ms	335.36 ms	738.84 ms	1,299.52 ms	2,017.4 ms
	Proposed	20.96 ms	41.92 ms	62.88 ms	83.84 ms	104.8 ms

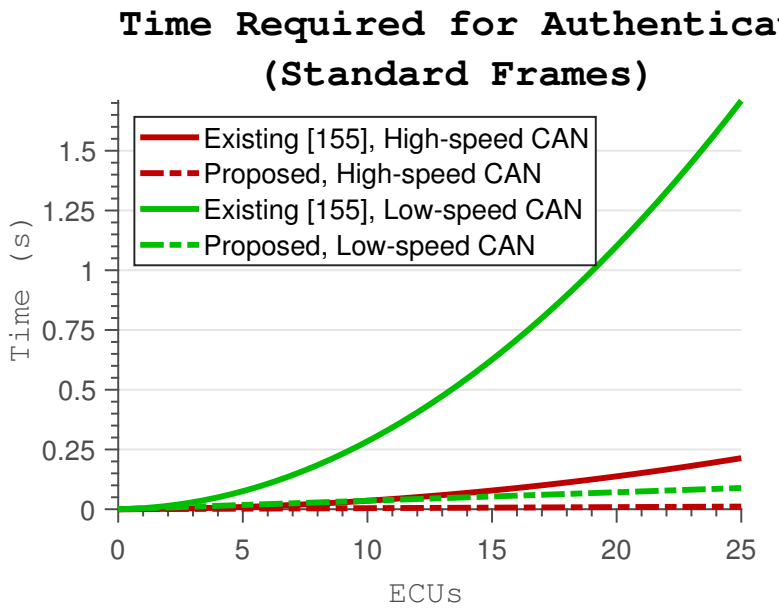


Figure 6.8: Authentication Phase Overhead Comparison (Standard Frames)

shared secrets by hashing them to 80-bit or 128-bit keys. In this way, the security of the encryption function is not reduced by the key generation.

6.7 Discussion and Concluding Remarks

In this chapter we present a novel CAN security framework based on PUF. The proposed framework offers improvements over previous PUF-based frameworks in terms of both scalability and the message overhead associated with normal operation. The savings in overhead results in our proposed framework being able to send the number of CAN frames required for the Authentication of a system with 20 nodes in only 6.5% of the time that it takes the existing framework. Normal message passing in our proposed framework requires only a single CAN frame to be sent while the existing approach requires two frames per message.

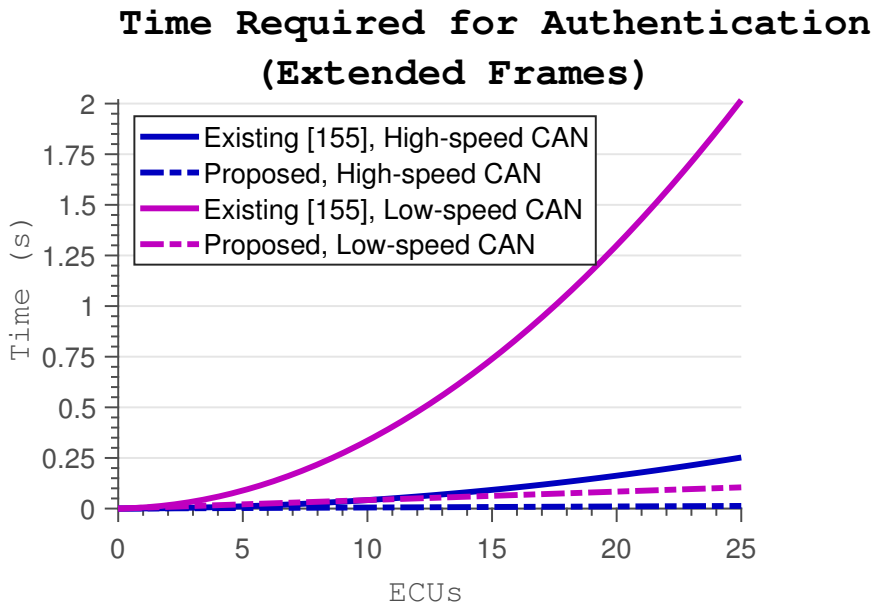


Figure 6.9: Authentication Phase Overhead Comparison (Extended Frames)

Our framework merely uses PRESENT and PHOTON as examples of LWC functions. PRESENT could be substituted for an alternative with a block size of 64-bits and a key size of at most 128-bits. PHOTON could be replaced by a different lightweight hash that is capable of producing a 128-bit output. Ongoing efforts in the development of LWCs, including NIST's efforts to standardize LWC, will likely result in new functions that offer better performance than what is currently available. Depending on the implementation focus, it might be preferable to choose an LWC that was optimized for hardware implementation rather than software implementation or vice-versa. One interesting avenue for future research would be a comprehensive study on the performance of various LWCs when implemented in both software for various resource-constrained platforms and in hardware such as FPGAs and ASICs.

Chapter 7

Conclusion and Future Work

Introducing security into IoT and vehicles is a difficult task without a clear and simple solution. In this work we have highlighted some of the proposed hardware methods for implementing security including the creation of hardware security modules and various applications for physically unclonable functions (PUFs). These applications involve methods for key storage, key generation, and message authentication. The focus of our research efforts has been to create PUFs specifically for cyber-physical systems such as IoT devices and vehicles. Our approach is to create PUFs from components that are commonly found in cyber-physical systems. Based on the observed reliability and uniformity values for the proposed PUF designs with respect to environmental conditions such as temperature, and in addition to the results of the uniqueness testing, this dissertation concludes that building PUFs within a cyber-physical system is a new direction worth exploring. This approach has the potential of being able to add security features to a device without needing to add any of the additional hardware that would normally be required to implement other security solutions. Similarly, this approach should help reduce the redesign costs that would be associated with adding security features to existing systems with known vulnerabilities. A singular PUF design would itself not be sufficient to make this approach a truly viable security approach for cyber-physical systems. The devices within these systems tend to have very limited and narrow roles. In order to carry out their required tasks, they would only require a similarly limited range of sensors and energy harvestors. Consider this information along with the fact that cyber-physical systems are used in a diverse range of fields including including medical devices, smart homes and cities, environmental monitoring, and industrial processes [27]. These facts make it unlikely that any singular sensor or energy harvester would be found consistently within cyber-physical systems across all of these fields. Thus, a PUF design based on a given component would be an option for a limited range of cyber-physical systems. It is for this reason that we proposed a number of designs based on different components which are all commonly found in cyber-physical systems.

The following contribution have been proposed in this work:

- Novel physically unclonable function (PUF) response balancing algorithm.
- Novel piezo sensor based PUF circuit design methodology.

- Novel thermistor temperature sensor based PUF circuit design methodology.
- Novel solar cell based PUF circuit design methodology.
- Evaluation of solar cell materials for the creation of novel PUF circuits.
- Novel PUF-Based Controller Area Network (CAN) security framework.

This work presents designs of physically unclonable functions (PUFs) which are specifically designed for use in cyber-physical systems. The majority of the contributions have already been reviewed by the scientific community and subsequently published in journals. The piezo PUF has also been a hardware demonstration at a conference. A provisional patent has been granted for the design methodology. The CAN security framework has been accepted for journal publication [97] and as a conference demonstration. From these developments we can conclude that there is merit behind our proposed methods of providing security to cyber-physical systems.

7.1 Future Work

The security of cyber-physical systems will continue to be a growing concern which will in term attract greater interest from researchers. The contributions of this work help to establish the viability of using PUFs created from sensors and energy harvesting devices as a security solution in cyber-physical systems. This dissertation has covered a number of important research topics, however, other areas still remain that require the attention of future research topics. The following items represent potential areas of future work that directly build upon those proposed in this dissertation:

- Evaluate the designs on a large population of physical devices. Our evaluations were performed on a limited number of prototypes or on simulated copies. A larger test population will provide a more accurate reference for what performance metrics could be expected of actual production copies of a design.
- Explore methods to allow energy harvesters to still function as a source of energy when the PUF is not in use. This would allow energy harvesters to truly be both a source of both power generation and security. Doing so should ease integration by reducing the need to add additional energy harvesters whose function would be to generate PUF responses.
- Explore methods for converting the design into a strong PUF. Strong PUFs have more potential applications than weak PUFs. However they are also known to be vulnerable to machine learning attacks [145] [173]. The computational component of the proposed designs could be leveraged to help safeguard against these attacks
- Investigate additional types of solar cell materials for their suitability in creating a PUF. Testing other solar cell materials would allow for the creation of a comprehensive catalog that could be used to determine if a solar cell based PUF is an appropriate security option for any solar powered system.

- Explore methods to improve the reliability of the proposed PUF designs. PUFs whose reliability value is less than 100% could produce erroneous responses. This is not ideal for security applications. Researchers have proposed error correcting codes (ECCs) and other schemes [33] [175] that could be used to improve the reliability values of the proposed designs.
- Explore methods to increase the tamper resistance of the proposed PUF designs [64] [103]. This characteristic was not a concern when creating the prototypes to test the validity of the proposed designs, but it would likely be one in a production quality implementation.
- Establish some computational performance benchmarks for the CAN security framework. This should include a selection of cryptosystems along with a selection of hardware implementations.

Bibliography

[1] Aishwarya, Farha Syed, Jaya Nupur, Aishwarya Vichare, and Arun Mishra. Authentication of electronic control unit using arbiter physical unclonable functions in modern automobiles. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ICTCS '16, pages 112:1–112:9, New York, NY, USA, 2016. ACM.

[2] AR Al-Ali, Imran A Zualkernan, Mohammed Rashid, Ragini Gupta, and Mazin Alikarar. A smart home energy management system using iot and big data analytics approach. *IEEE Transactions on Consumer Electronics*, 63(4):426–434, 2017.

[3] Mohammed Saeed Al-Kahtani. Survey on security attacks in vehicular ad hoc networks (vanets). In *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, pages 1–9. IEEE, 2012.

[4] Rafael Alvarez, Cándido Caballero-Gil, Juan Santonja, and Antonio Zamora. Algorithms for lightweight key exchange. *Sensors*, 17(7):1517, 2017.

[5] Muhammad N Aman, Kee Chaing Chua, and Biplab Sikdar. Position paper: Physical unclonable functions for iot security. In *Proceedings of the 2nd ACM international workshop on IoT privacy, trust, and security*, pages 10–13. ACM, 2016.

[6] Mani Amoozadeh, Arun Raghuramu, Chen-Nee Chuah, Dipak Ghosal, H Michael Zhang, Jeff Rowe, and Karl Levitt. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine*, 53(6):126–132, 2015.

[7] Ross Anderson and Markus Kuhn. Low cost attacks on tamper resistant devices. In *International Workshop on Security Protocols*, pages 125–136. Springer, 1997.

[8] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In *USENIX Security Symposium*, pages 1092–1110, 2017.

[9] AOSHIKE. Polycrystalline silicon solar cells. `www.amazon.com/AOSHIKE-Photovoltaic-Charger-Projects-54x54mm/dp/B07BLP238X`. Accessed: 2019-7-28.

[10] Erick Aponte. A study on energy harvesters for physical unclonable functions and random number generation. Master's thesis, Virginia Tech, 2017.

[11] Rosario Arjona, Miguel Prada-Delgado, Javier Arcenegui, and Iluminada Baturone. A puf-and biometric-based lightweight hardware solution to increase security at sensor nodes. *Sensors*, 18(8):2429, 2018.

[12] Aydin Aysu, Nahid Farhady Ghalaty, Zane Franklin, Moein Pahlavan Yali, and Patrick Schaumont. Digital fingerprints for low-cost platforms using mems sensors. In *Proceedings of the Workshop on Embedded Systems Security*, page 2. ACM, 2013.

[13] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[14] Elaine Barker and Quynh Dang. Nist special publication 800-57 part 1, revision 4. *NIST, Tech. Rep*, 2016.

[15] PSLM Barreto, Vincent Rijmen, et al. The whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*, volume 13, page 14, 2000.

[16] Chakib Bekara. Security issues and challenges for the iot-based smart grid. *Procedia Computer Science*, 34:532–537, 2014.

[17] Ygal Bendavid, Nasour Bagheri, Masoumeh Safkhani, and Samad Rostampour. Iot device security: Challenging "a lightweight rfid mutual authentication protocol based on physical unclonable function". *Sensors*, 18(12):4444, 2018.

[18] Elisa Bertino. Data security and privacy in the iot. In *Extending Database Technology (EDBT),19th International Conference on*, 2016.

[19] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, (2):76–79, 2017.

[20] Zoleikha Abdollahi Biron, Satadru Dey, and Pierluigi Pisu. Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):3893–3902, 2018.

[21] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 450–466. Springer, 2007.

[22] RR Brooks, S Sander, Juan Deng, and Joachim Taiber. Automobile security concerns. *IEEE Vehicular Technology Magazine*, 4(2), 2009.

[23] R. Buttigieg, M. Farrugia, and C. Meli. Security issues in controller area networks in automobiles. In *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 93–98, Dec 2017.

[24] B. Chatterjee, D. Das, S. Maity, and S. Sen. Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning. *IEEE Internet of Things Journal*, 6(1):388–398, Feb 2019.

[25] Baibhab Chatterjee, Debayan Das, Shovan Maity, and Shreyas Sen. Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning. *IEEE Internet of Things Journal*, 6(1):388–398, 2018.

[26] Bin Chen, Tanya Ignatenko, Frans MJ Willems, Roel Maes, Erik van der Sluis, and Georgios Selimis. High-rate error correction schemes for sram-pufs based on polar codes. *sign (L1· L2)*, 2:2, 2017.

[27] Hong Chen. Applications of cyber-physical system: a literature review. *Journal of Industrial Integration and Management*, 2(03):1750012, 2017.

[28] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao. Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017.

[29] Shuai Chen, Bing Li, and Yuan Cao. Intrinsic physical unclonable function (puf) sensors in commodity devices. *Sensors*, 19(11):2428, 2019.

[30] Kyong-Tak Cho and Kang G Shin. Fingerprinting electronic control units for vehicle intrusion detection. In *USENIX Security Symposium*, pages 911–927, 2016.

[31] B. Choi, S. Lee, J. Na, and J. Lee. Secure firmware validation and update for consumer devices in home networking. *IEEE Transactions on Consumer Electronics*, 62(1):39–44, February 2016.

[32] Wonsuk Choi, Hyo Jin Jo, Samuel Woo, Ji Young Chun, Jooyoung Park, and Dong Hoon Lee. Identifying ecus using inimitable characteristics of signals in controller area networks. *arXiv preprint arXiv:1607.00497*, 2016.

[33] Brice Colombier, Lilian Bossuet, Viktor Fischer, and David Hély. Key reconciliation protocols for error correction of silicon puf responses. *IEEE Transactions on Information Forensics and Security*, 12(8):1988–2002, 2017.

[34] MOST Cooperation. Most specification. *Rev. 3.0, URL: www. mostcooperation. com, Jun*, 2008.

[35] Craig Costello and Patrick Longa. Fourq: Four-dimensional decompositions on a q-curve over the mersenne prime. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, pages 214–235, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[36] Daniel Tudor Cotfas, Petru Adrian Cotfas, and Octavian Mihai Machidon. Study of temperature coefficients for parameters of photovoltaic cells. *International Journal of Photoenergy*, 2018, 2018.

[37] Jayita Das, Kevin Scott, Srinath Rajaram, Drew Burgett, and Sanjukta Bhanja. Mram puf: A novel geometry based magnetic puf with integrated cmos. *IEEE Transactions on Nanotechnology*, 14(3):436–443, 2015.

[38] PK Dash and NC Gupta. Effect of temperature on power output from different commercially available photovoltaic modules. *International Journal of Engineering Research and Applications*, 5(1):148–151, 2015.

[39] Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. Controller area network (can) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, Apr 2007.

[40] Jeroen Delvaux, Dawu Gu, Dries Schellekens, and Ingrid Verbauwhede. Helper data algorithms for puf-based key generation: Overview and analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(6):889–902, 2014.

[41] Jaya Dofe, Jonathan Frey, and Qiaoyan Yu. Hardware security assurance in emerging iot applications. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pages 2050–2053. IEEE, 2016.

[42] Shlomi Dolev, Łukasz Krzywiecki, Nisha Panwar, and Michael Segal. Optical puf for vehicles non-forwardable authentication. Technical report, Technical Report 15-02, Department of Computer Science, Ben-Gurion University of the Negev, 2015. Also appears as a Brief Announcement in IEEE NCA, 2015.

[43] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero. Vanet security surveys. *Computer Communications*, 44:1–13, 2014.

[44] Darrell Etherington and Kate Conger. Large ddos attacks cause outages at twitter, spotify, and other sites. `https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suffering-outage`. Access: 2019-02-08.

[45] Michael Feiri, Jonathan Petit, and Frank 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*, pages 9–18. ACM, 2013.

[46] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott. Emerging physical unclonable functions with nanotechnology. *IEEE Access*, 4:61–80, 2016.

[47] Blaise Laurent Patrick Gassend. Physical random functions. Master's thesis, Massachusetts Institute of Technology, 2003.

[48] Maria Gorlatova, John Sarik, Guy Grebla, Mina Cong, Ioannis Kymissis, and Gil Zussman. Movers and shakers: Kinetic energy harvesting for the internet of things. In *ACM SIGMETRICS Performance Evaluation Review*, volume 42, pages 407–419. ACM, 2014.

[49] Christopher Greer, Martin Burns, David Wollman, and Edward Griffor. Cyber-physical systems and internet of things. *NIST Special Publication*, 1900(202):52, 2019.

[50] Tech Briefs Media Group. `https://www.techbriefs.com/component/content/article/tb/supplements/mct/features/20190`. Accessed: 2019-8-7.

[51] Bogdan Groza, Pal-Stefan Murvay, Anthony Van Herrewege, and Ingrid Verbauwhede. Libra-can: A lightweight broadcast authentication protocol for controller area networks. In *CANS*, pages 185–200. Springer, 2012.

[52] Bogdan Groza and Stefan Murvay. Efficient protocols for secure broadcast in controller area networks. *IEEE Transactions on Industrial Informatics*, 9(4):2034–2042, 2013.

[53] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In *International workshop on Cryptographic Hardware and Embedded Systems*, pages 63–80. Springer, 2007.

[54] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[55] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet & Information Systems*, 8(12), 2014.

[56] Jian Guo, Thomas Peyrin, and Axel Poschmann. The photon family of lightweight hash functions. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 222–239, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[57] Hongmei He, Carsten Maple, Tim Watson, Ashutosh Tiwari, Jörn Mehnen, Yaochu Jin, and Bogdan Gabrys. The security challenges in the iot enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence. 2016.

[58] C. Herder, M. Yu, F. Koushanfar, and S. Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, Aug 2014.

[59] Daniel E Holcomb, Wayne P Burleson, and Kevin Fu. Power-up sram state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[60] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Security threats to automotive can networks–practical examples and selected short-term countermeasures. *Computer Safety, Reliability, and Security*, pages 235–248, 2008.

[61] Yohei Hori, Takahiro Yoshida, Toshihiro Katashita, and Akashi Satoh. Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas. In *Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on*, pages 298–303. IEEE, 2010.

[62] Zhao Huang and Quan Wang. A puf-based unified identity verification framework for secure iot hardware via device authentication. *World Wide Web*, 23(2):1057–1088, 2020.

[63] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017.

[64] Vincent Immler, Johannes Obermaier, Kuan Kuan Ng, Fei Xiang Ke, JinYu Lee, Yak Peng Lim, Wei Koon Oh, Keng Hoong Wee, and Georg Sigl. Secure physical enclosures from covers with tamper-resistance. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 51–96, 2019.

[65] Native Instruments. Controller area network (can) overview. *National Instruments, White paper*, 2014.

[66] Native Instruments. Flexray automotive communication bus overview. *National Instruments, White paper*, 2014.

[67] Texas Instruments. Tiva™ tm4c123gh6pm microcontroller-data sheet, 2013.

[68] ISO/IEC 29192-2:2019. Information technology – lightweight cryptography – part 2: Block ciphers. Standard, International Organization for Standardization, Geneva, CH, November 2019.

[69] ISO/IEC 29192-5:2016. Information technology – security techniques – lightweight cryptography – part 5: Hash-functions. Standard, International Organization for Standardization, Geneva, CH, August 2016.

[70] IXYS. Ixolar high efficiency solarbit kxob22-12x1f. `http://ixapps.ixys.com/DataSheet/KXOB22-12X1F_Nov16.pdf`, Nov. 2016. Accessed: 2019-7-28.

[71] Xin Jin, Asok Ray, and Robert M Edwards. Redundant sensor calibration and estimation for monitoring and control of nuclear power plants. *Trans. Amer. Nucl. Soc*, 101:307–308, 2009.

[72] Anju P Johnson, Rajat Subhra Chakraborty, and Debdeep Mukhopadhyay. A puf-enabled secure architecture for fpga-based iot applications. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):110–122, 2015.

[73] Anju P Johnson, Sayandeep Saha, Rajat Subhra Chakraborty, Debdeep Mukhopadhyay, and Sezer Gören. Fault attack on aes via hardware trojan insertion by dynamic partial reconfiguration of fpga over ethernet. In *Proceedings of the 9th Workshop on Embedded Systems Security*, page 1. ACM, 2014.

[74] H. Ju, Y. Kim, Y. Jeon, and J. Kim. Implementation of a hardware security chip for mobile devices. *IEEE Transactions on Consumer Electronics*, 61(4):500–506, November 2015.

[75] J Ju, Ray Chakraborty, Charles Lamech, and Jim Plusquellic. Stability analysis of a physical unclonable function based on metal resistance variations. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 143–150. IEEE, 2013.

[76] Jing Ju, Jim Plusquellic, Raj Chakraborty, and Reza Rad. Bit string analysis of physical unclonable functions based on resistance variations in metals and transistors. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, pages 13–20. IEEE, 2012.

[77] Deniz Karakoyunlu and Berk Sunar. Differential template attacks on puf enabled cryptographic devices. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010.

[78] Sye Loong Keoh, Sandeep S Kumar, and Hannes Tschofenig. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal*, 1(3):265–275, 2014.

[79] Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.

[80] Sajid Khan, Ambika Prasad Shah, Neha Gupta, Shailesh Singh Chouhan, Jai Gopal Pandey, and Santosh Kumar Vishvakarma. An ultra-low power, reconfigurable, aging resilient ro puf for iot applications. *Microelectronics Journal*, 92:104605, 2019.

[81] Mandeep Khera. Think like a hacker: Insights on the latest attack vectors (and security controls) for medical device applications. *Journal of diabetes science and technology*, 11(2):207–212, 2017.

[82] Z. King and Shucheng Yu. Investigating and securing communications in the controller area network (can). In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 814–818, Jan 2017.

[83] Pierre Kleberger, Tomas Olovsson, and Erland Jonsson. Security aspects of the in-vehicle network in the connected car. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 528–533. IEEE, 2011.

[84] Alicia Klinefelter, Nathan E Roberts, Yousef Shakhsheer, Patricia Gonzalez, Aatmesh Shrivastava, Abhishek Roy, Kyle Craig, Muhammad Faisal, James Boley, Seunghyun Oh, et al. 21.3 a 6.45 μw self-powered iot soc with integrated energy-harvesting power management and ulp asymmetric radios. In *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*, pages 1–3. IEEE, 2015.

[85] H Kohler. Most150–the next generation automotive infotainment backbone, smsc automotive infotainment systems. In *Proceedings of Workshop on ICT in Vehicles-PALEXPO, Geneva*, pages 5–7, 2008.

[86] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462, May 2010.

[87] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An intensive survey of fair non-repudiation protocols. *Computer communications*, 25(17):1606–1621, 2002.

[88] R Kumar and M Pallikonda Rajasekaran. An iot based patient monitoring system using raspberry pi. In *Computing Technologies and Intelligent Data Engineering (ICCTIDE), International Conference on*, pages 1–4. IEEE, 2016.

[89] S. Dinesh Kumar, Carson Labrado, Riasad Badhan, Himanshu Thapliyal, and Vijay Singh. Solar cell based physically unclonable function for cybersecurity in iot devices. In *2018 IEEE Computer Society Annual Symposium on VLSI, Hong Kong,*, pages 697–702. IEEE, July 2018.

[90] S Dinesh Kumar and Himanshu Thapliyal. Design of adiabatic logic-based energy-efficient and reliable puf for iot devices. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 16(3):1–18, 2020.

[91] Klaus Kursawe, Dries Schellekens, and Bart Preneel. Analyzing trusted platform communication. In *ECRYPT Workshop, CRASH-CRyptographic Advances in Secure Hardware*, 2005.

[92] C. Labrado and H. Thapliyal. Design of a piezoelectric-based physically unclonable function for iot security. *IEEE Internet of Things Journal*, 6(2):2770–2777, 2019.

[93] Carson Labrado, S Dinesh Kumar, Riasad Badhan, Himanshu Thapliyal, and Vijay Singh. Exploration of solar cell materials for developing novel pufs in cyber-physical systems. *SN Computer Science*, 1(6):1–13, 2020.

[94] Carson Labrado and Himanshu Thapliyal. Design of a piezoelectric based physically unclonable function for iot security. *IEEE Internet of Things Journal*, 2018.

[95] Carson Labrado and Himanshu Thapliyal. Hardware demo of a piezoelectric based PUF for hardware security in IoT devices. In *Hardware Oriented Security and Trust, 2019 IEEE International Symposium on*. IEEE, 2019.

[96] Carson Labrado and Himanshu Thapliyal. Hardware security primitives for vehicles. *IEEE Consumer Electronics Magazine*, 8(6):99–103, 2019.

[97] Carson Labrado, Himanshu Thapliyal, and Saraju Mohanty. Fortifying vehicular security through low overhead physically unclonable functions. *to appear ACM Journal of Emerging Technologies*.

[98] Carson Labrado, Himanshu Thapliyal, Stacy Prowell, and Teja Kuruganti. Use of thermistor temperature sensors for cyber-physical system security. *Sensors*, 19(18):3905, 2019.

[99] Jae W Lee, Daihyun Lim, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas 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*, pages 176–179. IEEE, 2004.

[100] Jungwon Lee, Seoyeon Choi, Dayoung Kim, Yunyoung Choi, and Wookyung Sun. A novel hardware security architecture for iot device: Pd-crp (puf database and challenge–response pair) bloom filter on memristor-based puf. *Applied Sciences*, 10(19):6692, 2020.

[101] Daihyun Lim, Jae W Lee, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, 2005.

[102] Chung-Wei Lin and Alberto Sangiovanni-Vincentelli. Cyber-security for the controller area network (can) communication protocol. In *Cyber Security (CyberSecurity), 2012 International Conference on*, pages 1–7. IEEE, 2012.

[103] Rui Liu, Huaqiang Wu, Yachun Pang, He Qian, and Shimeng Yu. A highly reliable and tamper-resistant rram puf: Design and experimental validation. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 13–18. IEEE, 2016.

[104] Xiaosen Liu and Edgar Sánchez-Sinencio. An 86% efficiency 12 μw self-sustaining pv energy harvesting system with hysteresis regulation and time-domain mppt for iot smart nodes. *IEEE Journal of Solid-State Circuits*, 50(6):1424–1437, 2015.

[105] Yu Liu, Kahin Akram Hassan, Magnus Karlsson, Ola Weister, and Shaofang Gong. Active plant wall for green indoor climate based on cloud and internet of things. *IEEE Access*, 6:33631–33644, 2018.

[106] Knud Lasse Lueth. State of the IoT 2018: Number of IoT devices now at 7B – market accelerating. `https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/`. Access: 2019-02-06.

[107] Hua-Dong Ma. Internet of things: Objectives and scientific challenges. *Journal of Computer science and Technology*, 26(6):919–924, 2011.

[108] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Intrinsic pufs from flip-flops on reconfigurable devices. In *3rd Benelux workshop on information and system security (WISSec 2008)*, volume 17, page 2008, 2008.

[109] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In *Towards Hardware-Intrinsic Security*, pages 3–37. Springer, 2010.

[110] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A systematic method to evaluate and compare the performance of physical unclonable functions. In *Embedded systems design with FPGAs*, pages 245–267. Springer, 2013.

[111] Rainer Makowitz and Christopher Temple. Flexray-a communication network for automotive control systems. In *2006 IEEE International Workshop on Factory Communication Systems*, pages 207–212, 2006.

[112] Cédric Marchand, Lilian Bossuet, Ugo Mureddu, Nathalie Bochard, Abdelkarim Cherkaoui, and Viktor Fischer. Implementation and characterization of a physical unclonable function for iot: a case study with the tero-puf. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):97–109, 2018.

[113] 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.

[114] Mcafee and the Center for Strategic and International Studies. 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.

[115] Kerry A McKay, Kerry A McKay, Larry Bassham, Meltem Sonmez Turan, and Nicky Mouha. *Report on lightweight cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2017.

[116] Mahammad Shareef Mekala and P Viswanathan. A novel technology for smart agriculture based on iot with cloud computing. In *I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on*, pages 75–82. IEEE, 2017.

[117] Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978.

[118] Dominik Merli, Frederic Stumpf, and Claudia Eckert. Improving the quality of ring oscillator pufs on fpgas. In *Proceedings of the 5th workshop on embedded systems security*, page 9. ACM, 2010.

[119] Charlie Miller and Chris Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015, 2015.

[120] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal. Pufchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (ioe). *IEEE Consumer Electronics Magazine*, 9(2):8–16, March 2020.

[121] M. R. Moore, R. A. Bridges, F. L. Combs, and A. L. Anderson. Data-driven extraction of vehicle states from can bus traffic for cyberprotection and safety. *IEEE Consumer Electronics Magazine*, 8(6):104–110, Nov 2019.

[122] Michael R Moore, Robert A Bridges, Frank L Combs, Michael S Starr, and Stacy J Prowell. Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, page 11. ACM, 2017.

[123] Arsalan Mosenia and Niraj K Jha. A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing*, 5(4):586–602, 2017.

[124] Debdeep Mukhopadhyay. Pufs as promising tools for security in internet of things. *IEEE Design & Test*, 33(3):103–115, 2016.

[125] Michael Müter and Naim Asaj. Entropy-based anomaly detection for in-vehicle networks. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 1110–1115. IEEE, 2011.

[126] Nicolas Navet and Françoise Simonot-Lion. In-vehicle communication networks-a historical perspective and review. Technical report, University of Luxembourg, 2013.

[127] Dmitry Nedospasov, Jean-Pierre Seifert, Clemens Helfmeier, and Christian Boit. Invasive puf analysis. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 30–38. IEEE, 2013.

[128] Thomas Nolte, Hans Hansson, and Lucia Lo Bello. Automotive communications-past, current and future. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, volume 1, pages 8–pp. IEEE, 2005.

[129] Yusuke Nozaki and Masaya Yoshikawa. Energy harvesting puf oriented id generation method and its evaluation system. In *Proceedings of the 2019 International Conference on Information Technology and Computer Communications*, ITCC 2019, pages 119–124, New York, NY, USA, 2019. ACM.

[130] Panasonic. Amorphous silicon solar cells. `https://panasonic.co.jp/ls/psam/en/products/pdf/Catalog_Amorton_ENG.pdf`, 2018. Accessed: 2019-7-28.

[131] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[132] S. Parkinson, P. Ward, K. Wilson, and J. Miller. Cyber threats facing autonomous and connected vehicles: Future challenges. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):2898–2915, 2017.

[133] Jonathan Petit, Christoph Bosch, Michael Feiri, and Frank Kargl. On the potential of puf for pseudonym generation in vehicular networks. In *Vehicular Networking Conference (VNC), 2012 IEEE*, pages 94–100. IEEE, 2012.

[134] PICASSO EU-US ICT Collaboration. Opportunity report towards enhanced EU-US ICT pre-competitive collaboration. 2012.

[135] Andreea-Ina Radu and Flavio D. Garcia. Leia: A lightweight authentication protocol for can. In Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows, editors, *Computer Security – ESORICS 2016*, pages 283–300. Springer International Publishing, 2016.

[136] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. *Journal of computer security*, 15(1):39–68, 2007.

[137] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[138] Garrett S Rose and Chauncey A Meade. Performance analysis of a memristive crossbar puf design. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[139] Kurt Rosenfeld, Efstratios Gavas, and Ramesh Karri. Sensor physical unclonable functions. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 112–117. IEEE, 2010.

[140] Matthew Ruff. Evolution of local interconnect network (lin) solutions. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 5, pages 3382–3389. IEEE, 2003.

[141] Ulrich Rührmair and Daniel E Holcomb. Pufs at a glance. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 347. European Design and Automation Association, 2014.

[142] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[143] Ryan A Scheel and Akhilesh 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*, pages 3–13. ACM, 2015.

[144] NXP Semiconductors. Kty81 series silicon temperature sensors, April 25, 2008.

[145] J. Shi, Y. Lu, and J. Zhang. Approximation attacks on strong pufs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10):2138–2151, 2020.

[146] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher clefia. In *International Workshop on Fast Software Encryption*, pages 181–195. Springer, 2007.

[147] Ali Shuja Siddiqui, Yutian Gui, Jim Plusquellic, and Fareena Saqib. A secure communication framework for ecus. *Advances in Science, Technology and Engineering Systems Journal*, 2(3):1307–1313, 2017.

[148] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–18, 2017.

[149] Measurement Specialties. Minisense 100 vibration sensor, May 12, 2009.

[150] ISO Standard. 11889. *Information Technology – Trusted Platform Module Library*, 2015.

[151] John A Stankovic. Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.

[152] Mile K Stojčev, Mirko R Kosanović, and Ljubiša 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*, pages 65–72. IEEE, 2009.

[153] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaâniche, and Youssef Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on*, pages 1–12. IEEE, 2013.

[154] Ying Su, Jeremy Holleman, and Brian 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*, pages 406–611. IEEE, 2007.

[155] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual design automation conference*, pages 9–14. ACM, 2007.

[156] Farha Syed, Jaya Nupur, Aishwarya Vichare, Arun Mishra, et al. Authentication of electronic control unit using arbiter physical unclonable functions in modern automobiles. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, page 112. ACM, 2016.

[157] Chris Szilagyi and Philip Koopman. Low cost multicast authentication via validity voting in time-triggered embedded control networks. In *Proceedings of the 5th Workshop on Embedded Systems Security*, page 10. ACM, 2010.

[158] Shahin Tajik, Enrico Dietz, Sven Frohmann, Helmar Dittrich, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, Christian Boit, and Heinz-Wilhelm Hübers. Photonic side-channel analysis of arbiter pufs. *Journal of Cryptology*, 30(2):550–571, 2017.

[159] Jack Tang, Ramesh Karri, and Jeyavijayan Rajendran. Securing pressure measurements using sensorpufs. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1330–1333. IEEE, 2016.

[160] Fatemeh Tehranipoor, Nima Karimian, Paul A Wortman, and John A Chandy. Low-cost authentication paradigm for consumer electronics within the internet of wearable fitness tracking applications. In *Consumer Electronics (ICCE), 2018 IEEE International Conference on*, pages 1–6. IEEE, 2018.

[161] Himanshu Thapliyal and Carson Labrado. Architecture for generating physically unclonable function response, Aug 2020. Application Number 16983329, (Provisional Patent).

[162] The National Institute of Standards and Technology (NIST) Computer Security Resource Center (CSRC). Lightweight cryptography. `https://csrc.nist.gov/projects/lightweight-cryptography`.

[163] Ramao Tiago Tiburski, Carlos Roberto Moratelli, Sergio F Johann, Marcelo Veiga Neves, Everton de Matos, Leonardo Albernaz Amaral, and Fabiano Hessel. Lightweight security architecture based on embedded virtualization and trust mechanisms for iot edge devices. *IEEE Communications Magazine*, 57(2):67–73, 2019.

[164] Shane Tuohy, Martin Glavin, Ciarán Hughes, Edward Jones, Mohan Trivedi, and Liam Kilmartin. Intra-vehicle networks: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):534–545, 2015.

[165] Parul Tyagi and Deepak Dembla. Investigating the security threats in vehicular ad hoc networks (vanets): Towards security engineering for safer on-road transportation. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 2084–2090. IEEE, 2014.

[166] M. Uddin, M. B. Majumder, and G. S. Rose. Robustness analysis of a memristive crossbar puf against modeling attacks. *IEEE Transactions on Nanotechnology*, 16:396–405, May 2017.

[167] Mesbah Uddin, Aysha S Shanta, Md Badruddoja Majumder, Md Sakib Hasan, and Garrett S Rose. Memristor crossbar puf based lightweight hardware security for iot. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2019.

[168] United States Computer Emergy Readiness Team (US-CERT). Alert (ta16-288a) heightened ddos threat posed by mirai and other botnets. `https://www.us-cert.gov/ncas/alerts/TA16-288A`. Access: 2019-02-08.

[169] Anthony Van Herrewege, Dave Singelee, and Ingrid Verbauwhede. Canauth-a simple, backward compatible broadcast authentication protocol for can bus. In *ECRYPT Workshop on Lightweight Cryptography*, volume 2011, 2011.

[170] S. Vergura and A. M. Pavan. On the photovoltaic explicit empirical model: Operations along the current-voltage curve. In *2015 International Conference on Clean Electrical Power (ICCEP)*, pages 99–104, 2015.

[171] Silvano Vergura. A complete and simplified datasheet-based model of pv cells in variable environmental conditions for circuit simulation. *Energies*, 9, 2016.

[172] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, et al. Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, 1(2011):9–52, 2011.

[173] A. Vijayakumar, V. C. Patil, C. B. Prado, and S. Kundu. Machine learning resistant strong puf: Possible or a pipe dream? In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 19–24, 2016.

[174] Weike Wang, Xiaobing Zhang, Qiang Hao, Zhun Zhang, Bin Xu, Haifeng Dong, Tongsheng Xia, and Xiang Wang. Hardware-enhanced protection for the runtime data security in embedded systems. *Electronics*, 8(1):52, 2019.

[175] Yuejiang Wen and Yingjie Lao. Efficient puf error correction through response weighting. In *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 849–852. IEEE, 2018.

[176] James B Wendt and Miodrag Potkonjak. Hardware obfuscation using puf-based logic. In *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*, pages 270–271. IEEE, 2014.

[177] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The internet of things—a survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

[178] Oliver Willers, Christopher Huth, Jorge Guajardo, and Helmut Seidel. Mems gyroscopes as physical unclonable functions. *IACR Cryptology ePrint Archive*, 2016:261, 2016.

[179] Marko Wolf and Timo Gendrullis. Design, implementation, and evaluation of a vehicular hardware security module. In *International Conference on Information Security and Cryptology*, pages 302–318. Springer, 2011.

[180] Marko Wolf, André Weimerskirch, and Christof Paar. Secure in-vehicle communication. *Embedded Security in Cars*, pages 95–109, 2006.

[181] S. Woo, H. J. Jo, and D. H. Lee. A practical wireless attack on the connected car and security protocol for in-vehicle can. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):993–1006, April 2015.

[182] Y. Wu, Yeon-Jin Kim, Zheyan Piao, J. Chung, and Yong-En Kim. Security protocol for controller area network using ecandc compression algorithm. In *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–4, Aug 2016.

[183] Peter Würfel. *Physics of solar cells*, volume 1. Wiley-vch Weinheim, 2005.

[184] Teng Xu, James B Wendt, and Miodrag Potkonjak. Security of iot systems: Design challenges and opportunities. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, pages 417–423. IEEE Press, 2014.

[185] Teng Xu, James Bradley Wendt, and Miodrag Potkonjak. Secure remote sensing and communication using digital pufs. In *Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems*, pages 173–184. ACM, 2014.

[186] Eli Yablonovitch, Owen D Miller, and SR Kurtz. The opto-electronic physics that broke the efficiency limit in solar cells. In *Photovoltaic Specialists Conference (PVSC), 2012 38th IEEE*, pages 001556–001559. IEEE, 2012.

[187] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal. Pmsec: Physical unclonable function-based robust and lightweight authentication in the internet of medical things. *IEEE Transactions on Consumer Electronics*, 65(3):388–397, Aug 2019.

[188] Abel Yeboah-ofori, Jamal-Deen Abdulai, and Ferdinand Katsriku. Cybercrime and risks for cyber physical systems: A review. 2018.

[189] Saman Taghavi Zargar, James Joshi, and David Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069, 2013.

[190] Ji-Liang Zhang, Gang Qu, Yong-Qiang Lv, and Qiang Zhou. A survey on silicon pufs and recent advances in ring oscillator pufs. *Journal of computer science and technology*, 29(4):664–678, 2014.

[191] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.

Vita

EDUCATION

2017 Master of Science in Electrical Engineering
University of Kentucky, Lexington, KY
2014 Bachelor of Science in Electrical Engineering
University of Kentucky, Lexington, KY
2014 Bachelor of Science in Computer Engineering
University of Kentucky, Lexington, KY

HONORS AND AWARDS

2019 Dean's Award for Outstanding Teaching Assistant
University of Kentucky College of Engineering

PUBLICATIONS and PATENTS

1. H. Thapliyal and C. Labrado, "Architecture for generating physically unclonable function response", Provisional Patent. Application Number 16983329, Filed August 3, 2020.

2. C. Labrado, H. Thapliyal and S. Mohanty, "Fortifying Vehicular Security Through Low Overhead Physically Unclonable Functions". ACM Journal of Emerging Technologies in Computing Systems, 2021 (Accepted for Publication).

3. C. Labrado, S. D. Kumar, R. Badhan, H. Thapliyal, and V. Singh, "Exploration of Solar Cell Materials for Developing Novel PUFs in Cyber-Physical Systems". SN COMPUT. SCI. 1, 313 (2020).

4. T. Cultice, C. Labrado and H. Thapliyal, "A PUF Based CAN Security Framework," 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Limassol, Cyprus, 2020, pp. 602-603.

5. C. Labrado and H. Thapliyal, "Hardware Security Primitives for Vehicles," in IEEE Consumer Electronics Magazine, vol. 8, no. 6, pp. 99-103, 1 Nov. 2019.

6. C. Labrado, H. Thapliyal, S. Prowell, and T. Kuruganti, "Use of Thermistor Temperature Sensors for Cyber-Physical System Security". Sensors 2019, 19, 3905.

7. H. Thapliyal, N. Ratajczak, O. Wendroth and C. Labrado, "Amazon Echo Enabled IoT Home Security System for Smart Home Environment," 2018 IEEE International

Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Hyderabad, India, 2018, pp. 31-36.

8. C. Labrado and H. Thapliyal, "Design of a Piezoelectric-Based Physically Unclonable Function for IoT Security," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2770-2777, April 2019.

9. S. D. Kumar, C. Labrado, R. Badhan, H. Thapliyal and V. Singh, "Solar Cell Based Physically Unclonable Function for Cybersecurity in IoT Devices," 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, 2018, pp. 697-702.

10. H. Thapliyal, V. Khalus and C. Labrado, "Stress Detection and Management: A Survey of Wearable Smart Health Devices," in IEEE Consumer Electronics Magazine, vol. 6, no. 4, pp. 64-69, Oct. 2017.

11. C. Labrado, H. Thapliyal and F. Lombardi, "Design of majority logic based approximate arithmetic circuits," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

12. C. Labrado and H. Thapliyal, "Design of a multilayer five-input majority gate and adder/subtractor circuits in NML computing," in Electronics Letters, vol. 52, no. 19, pp. 1618-1620, 15 9 2016.

13. C. Labrado and H. Thapliyal, "Design of adder and subtractor circuits in majority logic-based field-coupled QCA nanocomputing," in Electronics Letters, vol. 52, no. 6, pp. 464-466, 17 3 2016.

14. H. Thapliyal, C. Labrado, and K. Chen, "Design procedures and NML cost analysis of reversible barrel shifters optimizing garbage and ancilla lines". J Supercomput 72, 1092–1124 (2016).

15. C. Labrado, H. Thapliyal and R. F. Demara, "Design of Testable Adder Circuits for Spintronics Based Nanomagnetic Computing," 2015 IEEE International Symposium on Nanoelectronic and Information Systems, Indore, 2015, pp. 107-111.

Demonstrations

1. C. Labrado and H. Thapliyal, "Hardware Demo of Thermistor and Solar Cell Based PUFs via a PUF based Controller Area Network Security Framework", Accepted as hardware demonstration to appear at 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST).

2. C. Labrado and H. Thapliyal, "Hardware Demo of a Piezoelectric Based PUF for Hardware Security in IoT Devices", Hardware demonstration presented at 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST).