AN ICMETRIC BASED MULTIPARTY COMMUNICATION FRAMEWORK

Hasan Tahir

THOUGHT THE HARDER · HEART THE KEENER

A thesis submitted for the degree of
Doctor of Philosophy
at
School of Computer Science and Electronic Engineering
University of Essex
2017

Dedicated to Mama and Papa

ABSTRACT

Cryptographic algorithms have always relied on stored keys for the provision of security services. Since these keys are stored on a system this makes them prone to attack. Efforts to increase the key size makes brute forcing difficult but does not eliminate key theft.

This thesis proposes a comprehensive security framework for groups of devices. The research makes four major contributions to improve the security of devices in the multiparty environment. The proposed framework uses the novel Integrated Circuit Metric (ICMetric) technology which proposes utilizing measurable properties and features of a device to create a device identification. This device identification called the ICMetric is used to create cryptographic keys which are then used in the designed cryptosystems.

The first contribution of the thesis is the creation of an ICMetric using sensors found in modern smart devices. The research explores both explicit and implicit features which can be used to generate of an ICMetric.

The second contribution of this research is the creation of a group ICMetric which is computed using the device ICMetric. The computation of the device ICMetric is a particular challenge as it has to be computed without violating the properties of the ICMetric technology.

The third contribution is the demonstration that an ICMetric can be used for the creation of symmetric key. The fourth contribution of this research is an efficient RSA based asymmetric key generation scheme for the multiparty environment.

Designing a system using widely accepted cryptographic primitives does not guarantee a secure system therefore the security of proposed schemes has been studied under the standard model. The schemes presented in this thesis attempt to improve the security of devices in the group environment. The schemes demonstrate that key theft deterrent technologies can be incorporated into cryptographic schemes to offer higher levels of security and privacy.

ACKNOWLEDGEMENTS

First of all I would like to thank Allah, the Almighty for giving me the ability and strength to carry out this research.

My deepest gratitude to my supervisor Prof. Klaus McDonald-Maier for his continuous support during the Ph.D study. I could not have imagined having a better supervisor and mentor for my Ph.D. A heartfelt thank you to my PhD board chair Prof. Stuart Walker for his valuable feedback and thought provoking questions. I am grateful for the University of Essex Doctoral Scholarship Award and the School of Computer Science and Electronic Engineering for making it possible for me to study here.

I would like to thank my sister Ruhma Tahir for her support during the PhD. Completing this PhD would not have been possible without her help. I also thank my fellow lab mate Khattab M Ali for being a sincere friend.

I must thank my family: parents and brother for their encouragement and prayers throughout the research. I would also thank my wife for being understanding during even the toughest times of my research.

Finally, I must mention my daughter Anabia, nephew Zaim and niece Eesha who's smiles have been a source of joy for me even when I was stressed.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AES	Advanced Encryption Standard
CSPRNG	Cryptographically Secure Pseudo Random Number Generator
DoS	Denial of Service
HMAC	Hashed Message Authentication Code
ICMetric	Integrated Circuit Metric
IoT	Internet of Things
IV	Initialization Vector
MAC	Media Access Control Address
NIST	National Institute of Standards and Technology
PBKDF	Password Based Key Derivation Function
PKC	Public Key Cryptography
PUF	Physically Unclonable Function
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman (public key encryption technology)
SHA	Simple Hash Algorithm
WSN	Wireless Sensor Network

LIST OF PUBLICATIONS

[1] H. Tahir, R. Tahir, K. McDonald-Maier, ”On the Security of Consumer Wearable Devices in the Internet of Things”, Undergoing revisions for publication in PLOS One.

[2] R. Tahir, H. Tahir, A. Sajjad, K. McDonald-Maier, ”A Secure Cloud Framework for ICMetric Based IoT Health Devices”, Accepted for publication in 2nd International Conference on Internet of Things, Data and Cloud Computing (ICC 2017), Cambridge, UK, 22-23 March, 2017.

[3] H. Tahir, R. Tahir, K. McDonald-Maier, A. Fernando, ”A Novel ICMetric Based Framework for Securing the Internet of Things”, IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 7-11 January 2016, pp. 469-470.

[4] R. Tahir, H. Tahir, K. McDonald-Maier, “Securing Health Sensing Using Integrated Circuit Metric”, MDPI Sensors Journal 2015, October, 2015, 15(10), pp 26621-26642.

[5] H. Tahir, R. Tahir, K. McDonald-Maier, “Securing MEMS Based Sensor Nodes in the Internet of Things”, 6th International Conference on Emerging Security Technologies (EST), Technische Universitaet Braunschweig, Germany, 3-5 September, 2015.

[6] H. Tahir, G. Howells, H. Hu, D. Gu, K. McDonald-Maier, “On the Incorporation of Secure Filter in ICMetrics Group Communications”, 5th International conference on Emerging Security Technologies (EST), Alcala de Henares, Spain, 10-12 September, 2014.

[7] H. Tahir, G. Howells, H. Hu, D. Gu, K. McDonald-Maier, “On Secure Group Admission Control Using ICMetrics”, 5th International conference on Emerging Security Technologies (EST), Alcala de Henares, Spain 10-12 September, 2014.

[8] H. Tahir, R. Tahir, K. McDonald-Maier, “A Group Secure Key Generation and Transfer Protocol Based on ICMetrics”, 9th IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP 14), Manchester, UK, 23-25 July, 2014. pp 733-738.

[9] H. Tahir, R. Tahir, K. McDonald-Maier, “A Novel Private Cloud Document Archival System Based on ICMetrics”, 4th International conference on Emerging Security Technologies (EST), Cambridge, 09-11 Sept, 2013. pp 102-106.

CHAPTER 1

INTRODUCTION

Recently, there has been a visible reduction in the cost of computing which has resulted in the creation of new venues that utilize computers in their various forms. In their early years, computers were standalone devices which could process data when it was provided to them. Emergence of energy efficient radio technologies resulted in the creation of autonomous communication systems like Wireless Sensor Networks (WSN's) [1][2]. WSN's relied on using leaf nodes which could sense a stimulus and forward the information to the base node [3]. After the emergence of WSN's, research was geared towards providing security to the sensor node, data and communications of the network. WSN's were primarily designed to operate as a standalone network of sensors. The emergence of internet and smart devices caused WSN's to evolve into more intelligent environments where devices could sense and communicate what is happening around them and then forward that information via the internet. The Internet of Things [4] is one such environment which aims to create a collaboration between devices for generating, processing and sharing data. The popularity of Internet of Things and the general appeal of collaborative environments means that in the future, devices will be increasingly functioning in a group and less as standalone devices. Many of these devices will sense, process, store and communicate data of sensitive nature.

Secrecy of sensitive data is an important requirement especially when the data is being communicated beyond the confines of the device. The Cambridge dictionary [5] defines cryptography as the practice of creating and understanding codes that keep information secret. Ensuring secrecy and privacy has become a complex task as adversaries possess resources and the capability of exploiting weakness in a system. A fundamental problem in cryptography is how to communicate securely in the presence of adversaries. This problem has become even more important with the proliferation of ubiquitous smart devices, Internet of Things and group communications. Often adversaries exploit design weaknesses to gain illegitimate access to a system. Such types of attacks can be difficult to correct as a solution may lie in redesigning of the system. A problem that has plagued the field of cryptography is key theft. An attack on the cryptographic keys can be sufficient to compromise the system.

1.1 RESEARCH MOTIVATION

Today there are more devices than the number of users on the internet. This means many of us have more than one device which connects us to the internet. As everyday objects like televisions and watches become internet capable the importance of security cannot be denied. Despite constant research in the field of cryptography, computation devices are still insecure. Numerous computer security incidents are reported every year in which systems are attacked resulting in financial loss, data theft and even threat to life [6]. With easy availability of computation power and increased connectivity adversaries are now stronger than ever before. For an adversary the motivation behind an attack could be to compromise national security or to just create a low level nuisance.

Emergence of high capacity networks has created many new applications like teleconferencing, real-time information services and collaborative environments which follow the group communication model. In this model one or more authorized senders send messages to one or more authorized receivers. Numerous devices are already being marketed for the Internet of Things environment. The devices function collaboratively to enable the sharing of data and information. A recent detailed study [7] shows that devices in the Internet of Things are insecure and often lack the resources required for the provision of security. To an adversary the group communication presents an attractive environment which is abundant in devices and communication links. Hence adversaries will attempt to gain access by exploiting flaws in security or system design. Security schemes and protocols that require a cryptographic key assume that the key is kept secret. This assumption alone is a weakness in any cryptosystem as there are many ways for an adversary to capture the keys. If an unencrypted cryptographic key is captured then the security of the system is compromised. Thus cryptographic key theft is an Achilles Heel for any security based system. The fact that an attack on cryptographic keys can lead to failed security creates the impending case for a renewed approach for the provision of security in group environments.

1.2 THESIS STATEMENT

Cryptographic schemes are based on publically available protocols and algorithms while the security keys are kept secret. Hence the security of a system lies in keeping the keys secret and not the underlying protocol. If at any point the keys are captured then the system can be compromised. A password stored in the human’s memory is secure and cannot be unwillingly known to any other person. On the contrary, cryptographic keys are stored on the system which makes them prone to both internal and external attacks. Cryptographers increase the key size to make it difficult for an adversary to brute force a cryptographic key [8]. Increasing the key entropy as a method of deterring key theft is impractical as there are numerous methods of capturing keys [6][9][10].

Cryptographic key theft in a group setting creates a unique environment where there are multiple attractive targets which an attacker can capture. Once a single system is compromised in a group then the attack can be escalated to capture the entire group. Owing to the existence of multiple points of attacks, group communications are at greater risk of being attacked. Similarly, when a single message is communicated in the group environment, it traverses through a large number of links which increases the possibility of the message being intercepted by an adversary.

This research studies a comprehensive framework that provides security to devices communicating in a group setting. The aim of this research is to put into practice theories and concepts of the Integrated Circuit Metric (ICMetric) technology in a multiparty environment. The research demonstrates that it is possible to use the features of a device to create an identification that provides security in the group setting. To achieve this the research investigates possible properties and features which can be used for the creation of an ICMetric of a device. The study aims to show evidence that the ICMetric technology can be used to generate cryptographic keys for the provision of authentication, confidentiality and integrity. The presented framework studies the ICMetric technology in two ways one: as a method of key theft deterrence and two: as a basis for cryptographic key generation.

1.3 THESIS CONTRIBUTIONS

This thesis investigates a set of topics to ensure security of devices communicating in a multiparty environment. The contributions of this thesis provide a means for enabling high levels of security in devices that perform communications and computations in a group environment. This thesis provides a secure framework which can be adapted to any environment where there are devices that function collaboratively. Conventionally, cryptography has relied on stored keys for the provision of security. Stored keys are considered a vulnerability as they can be captured by an adversary. Hence by incorporating a key theft deterrent like the ICMetric technology provides a way of mitigating weaknesses that plague the field of cryptography.

This thesis demonstrates that unique features of a device can be used to provide an identity to a devices which can then be used for the provision of security services. Hence, the first contribution of this thesis is that it explores unique explicit and implicit features which can be used to generate a device identity called the ICMetric. The ICMetric is a unique property which is why it cannot be communicated or stored on the system. This forms a challenge since a group ICMetric needs to be generated to identify devices communicating together in a group.

The second contribution of this thesis is the provision of a scheme that assists in the generation of a group ICMetric while preserving the properties of the ICMetric. The research shows that the group ICMetric can be used for the creation of cryptographic keys for the group.

The third contribution of this thesis is the creation of a symmetric key for the group by using the group ICMetric. The symmetric key generation algorithm relies on using the group ICMetric, well established security primitives and algorithms to create a symmetric key for the group. The scheme is also composed of an authentication method that facilitates ICMetric based authentication.

The fourth contribution of this thesis is the creation of a scheme that uses the group ICMetric and RSA algorithm to generate asymmetric keys for the group.

Perhaps the greatest contribution of this thesis is that it delivers high levels of security without having to make drastic changes to existing security systems. Thus the ICMetric technology can be integrated into any computation system with minimum impact on existing infrastructure or technology. The ICMetric technology has been designed to integrate with conventional systems as it has been designed as a distinct add-on layer.

1.4 SECURITY AIMS

At the heart of a cryptosystem are goals around which development takes place. Each security goal must work in harmony with other security goals so that the resulting system is fully secure. Security goals are central to this research as system design choices are made based on the selected security goals. The security framework proposed in this research aims to fulfil three basic security goals i.e. authentication, confidentiality and integrity. Given below are the security goals of the project and how they can be interpreted with reference to the problem statement.

- **Authentication** – provide systems with an identity and verify the correctness of the identity (machine authentication).
    - **Access Control** – limit access to only authenticated entities. Thus block unwanted or illegitimate access.
- **Confidentiality** – ensure that communications are accessible to only authenticated entities.
- **Integrity** – in this security goals the aim is to ensure the purity and trustworthiness of communications. Through integrity the system prevents unauthorized systems from making contributions and modifications to communications. Also prevent authorized participants from making dishonest contributions and modifications to communications.

1.5 PHYSICAL ROOT OF TRUST

Adversaries now have access to high power computing with sufficient resources to conduct a powerful attack. Therefore, researchers explore other supporting methods which can enhance the security of conventional cryptography. Traditionally, cryptographic systems have relied on mathematical intractability [11] of primitives to guarantee security. Mathematical intractability is not sufficient to secure a system as often an adversary will not behave as expected. An adversary will often employ methods which do not exploit mathematical intractability like side channel and cold boot to attack a system. Therefore, a new breed of methods and primitives are required that are based on physical reasoning [12]. The use of physical reasoning to build a cryptographic system can ensure higher levels of security because the system primitives are rooted in the physical world. The physical root of trust in this research is the Integrated Circuit Metric technology. The ICMetric technology uses physical features of a device for the provision of cryptographic services. Figure 1.1 shows the physical world and its connection to the physical root of trust i.e. ICMetric technology. The physical root of trust is used to provide a basis for cryptographic primitives. The cryptographic primitives form building blocks for a set of security goals required for a secure multiparty environment.

A concept similar to ICMetric is physically unclonable functions which are also used to provide hardware entangled security. A physically unclonable function uses a challenge-response system as a unique identifying feature. Hence a unique response to an input challenge is used to design systems based on the physically unclonable function. The ICMetric technology does not rely on a challenge-response system to create a device identification. Instead features are directly accessed and processed to provide security. What sets the ICMetric technology apart from physically unclonable functions is that the ICMetric technology uses multiple device features (rather than one) and processes them as a foundation for a range of services. The ICMetric technology can be used for providing security services like authentication, key generation, confidentiality etc.

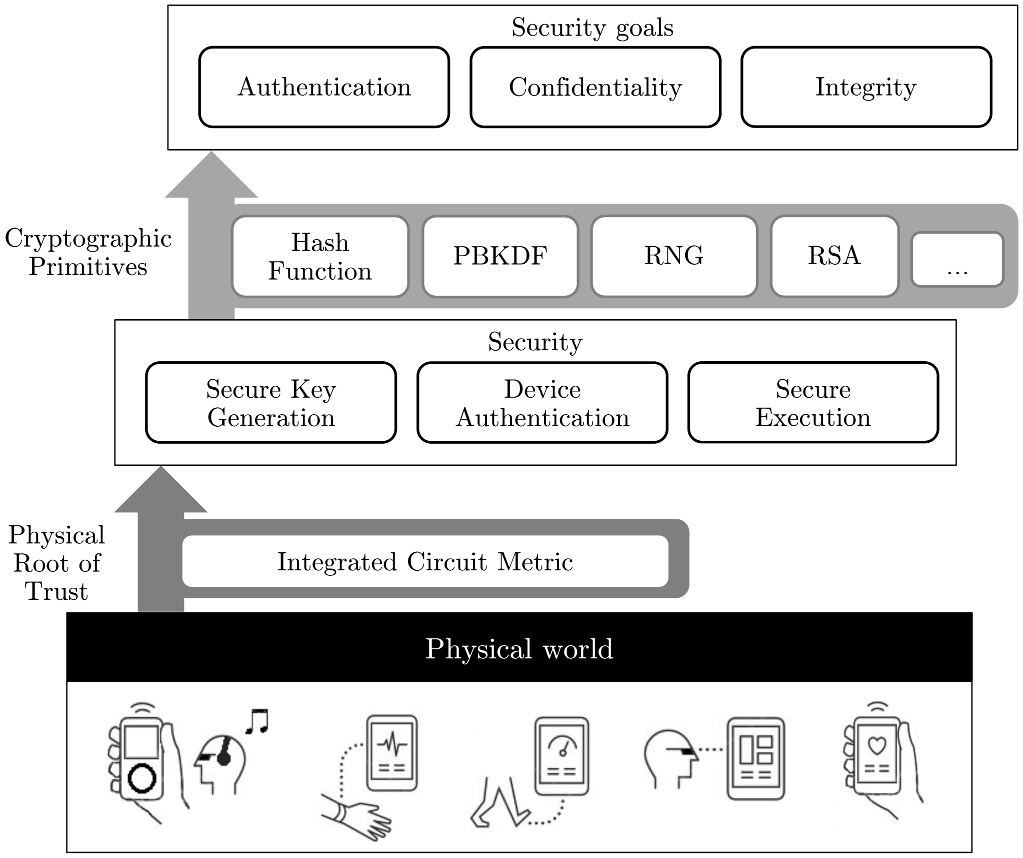

Figure 1.1. Relationship between the physical world, root of trust and cryptography

1.6 THESIS STRUCTURE

In response to the challenges faced by devices in group environment this thesis presents a comprehensive security framework that is based on the ICMetric technology. The contributions are arranged into chapters as follows:

- Chapter 2 focuses on literature related to group environment and the security of devices in the group. The chapter begins with the description of a communication suite for devices in the Internet of Things. The suite is revisited so that it encompasses the ICMetric technology. The chapter also introduces security concerns that need to be addressed when securing devices in a group environment.
- Chapter 3 introduces the ICMetric technology as a physical root of trust. The chapter explores the concept of a bias in a MEMS accelerometer,

gyroscope and strain gauge sensors. The chapter explores implicit and explicit features of a wearable health sensor for ICMetric generation. A detailed statistical study has also been presented for each MEMS sensor. The statistical study shows that each sensor possesses sufficient bias which can be used for ICMetric generation. After the creation of individual ICMetric the chapter provides a detailed account of how a group ICMetric can be generated using the ICMetric of individual devices.

- In chapter 4 the creation of a symmetric key for groups of devices is explored. The chapter explores using Password Based Key Derivation Function to create a symmetric key for the group. This function is based on security primitives like salting, hashing and a large iteration count. The chapter explores these primitives and presents a novel algorithm that uses the group ICMetric to create a symmetric key for the multiparty environment. The chapter concludes with a performance analysis of the proposed symmetric key generation algorithm.
- Chapter 5 focuses on the creation of an asymmetric key for the group environment. The proposed scheme uses the RSA algorithm with a group ICMetric to create an asymmetric key for the group. The chapter concludes with a performance analysis of the proposed asymmetric key generation algorithm.
- Chapter 6 presents a security analysis of the proposed schemes in the standard model. Security proofs have been designed that test the proposed schemes by deliberately placing adversaries while various entities interact. The security proofs prove the security of the ICMetric technology, symmetric key generation, asymmetric key generation and prominent scheme primitives.
- Chapter 7 closes the thesis with a conclusion and provides directions which can be explored for future research.

CHAPTER 2

LITERATURE REVIEW

As computing devices became ubiquitous the next logical step in the evolution of computers was to enable interconnectivity of devices for data and information sharing. The interconnectivity of devices is not a new concept, but it has come under renewed spotlight after the emergence of high capacity networks and small sized devices. Whenever devices share data and resources, it is important that both the communicating and the communicated are secure from attacks. This chapter presents a bird's eye view of earlier works in the field of multiparty communications. A discussion on multiparty communications is incomplete without referring to the internet of things and the many devices that form part of the internet of things environment. Since internet of things is an emerging field of research therefore a survey of possible attacks on devices in the internet of things has been presented with focus on wearable technologies. This chapter highlights two recent security advancements i.e. physically unclonable functions and device fingerprinting. These two areas of secure computing form the basis of the ICMetric technology. The chapter explains the design principles of the ICMetric technology and how the features of a device can be used to form an identification of a device which is then used for the provision of cryptographic services.

2.1 INTERNET OF THINGS

The Internet of Things (IoT) is a network of physical devices which collect and exchange data through the many form of network connectivity [13][14]. Thus the IoT is composed of multiple smart devices which can sense and communicate. The smart devices are intended to be worn on the body, carried by the owner, fitted on a wall or even installed ubiquitously. Hence the IoT presents a unique environment where devices with varying capability and resources are generating and sharing data. The emergence of IoT is a result of creating devices which are interoperable, thus they can share data and information.

What sets IoT apart from regular computer networks is the fact that devices in the IoT ecosystem are sensors, devices, objects which are not considered computers. IoT devices are intended to be ubiquitous devices that function with minimum user intervention. Hence IoT devices are both consumers and producers of data. Broadly the IoT is the result of convergence of a number of technological trends as follows:

- Ubiquitous computing – the creation of smart technology by embedding microprocessors in everyday objects so they can communicate and sense their surroundings.
- Universal internet connectivity – the use of IP based networking to facilitate data and information sharing.
- Miniaturization of technology – the reduction in size of computing technology owing to circuit miniaturization, microprocessors and embedded systems.
- Cloud computing – the rise of cloud technology which enables resource sharing and also allows analytical feature aggregation via the cloud.

- Data analytics – algorithms and processes that facilitate the creation of knowledge from raw data obtained via the cloud.

Based on a recent research [15] the IoT can be depicted as a multitier architecture with four layers namely perceptual layer, network layer, support layer and application layer. The perceptual layer is closest to the physical world while the top layers address issues related to data processing and information retrieval. The perceptual layer is composed of sensors and devices that interact with the physical world. The devices share data with the network layer so that data can be communicated across various networks. The abstraction layer supports the extraction of information from incoming datasets. The abstraction layer presents data to the application layer for customized information services. The application layer processes and presents the data provided by the abstraction layer. Figure 2.1 is the four layer communication suite for the IoT.

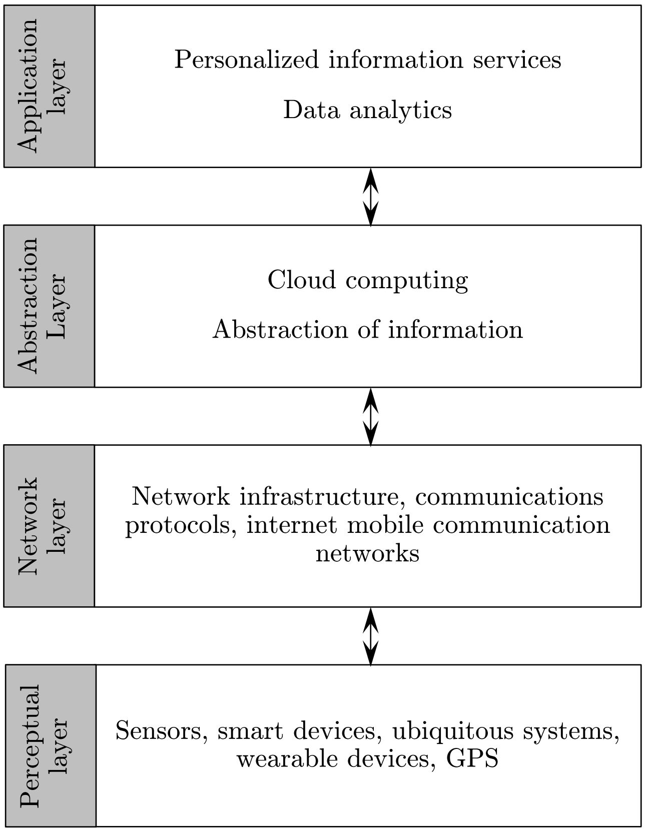
Application layer
Personalized information services
Data analytics
Abstraction Layer
Cloud computing
Abstraction of information
Network layer
Network infrastructure, communications protocols, internet mobile communication networks
Perceptual layer
Sensors, smart devices, ubiquitous systems, wearable devices, GPS

Figure 2.1. A four layer IoT communication suite [15]

2.1 INTERNET OF THINGS

IoT is rapidly penetrating a wide range of domains for instance health monitoring, home automation, lifestyle, fitness monitoring, industrial support, entertainment, gaming etc. Below is a brief description of device categories in the IoT.

- Health monitoring – wearable devices that measure physiological signals of its wearer. These devices can take body readings and also help with fitness monitoring. These devices measure heart rate, steps taken, distance covered and calories burnt during a physical exercise.
- Smart Utilities and home automation – the devices in this category automate the home by providing remote appliance control, home intrusion detection and smart metering.
- Industrial support – devices intended to be worn in an industrial environment. Devices in this category help with a large range of tasks like logistics, hazard monitoring, indoor asset location determination, process automation activities and ecommerce [16].
- Entertainment – wearable devices that can stream audio and video. The devices in this category can be wireless headphones, speakers, and wearable displays with the ability of connecting to wide range of entertainment systems. Some devices are also used to create immersive environment during gameplay.
- Lifestyle – general purpose wearable devices that provide internet, cellular and other forms of connectivity. These devices make it convenient for the user to carry out their everyday activities. It is these and many other devices which will integrate to form the smart cities [17].

Manufacturers of IoT devices are eager to capture an emerging market therefore the first devices to emerge were IoT capable versions of devices which we use every day. Research has been done on designing IoT capable wearable technology. Internet capable watches, fitness trackers, fitness bands are just some of the few products which are a result of rapid research in the field of wearables in the IoT. Even though manufacturers have been successful in rapidly designing devices for the IoT. They have done so at the cost of lack of necessary services like security [18]. Wearable technologies make many promises but also possesses barriers [19] in their adoption. To unlock the full potential of IoT it is necessary that the devices possess both resources and the ability to provide security services.

2.2 SECURITY CONCERNS

Adversaries are often able to exploit weaknesses in a system to gain illegitimate access. As systems move out from the security of homes and offices to more ubiquitous settings, the importance of security cannot be denied. It is important to ensure the security of both hardware and software components of any system. Given below is a discussion on possible system attacks and their prevalence in everyday life.

2.2.1 Physical Attacks

A security concern with any hardware device is physical tampering. Since hardware devices process and store data therefore it is important to protect hardware from attacks which could lead to data being captured or modified. Conventionally, data processing is limited to the devices embedded system and external access is prohibited to defeat device tampering. Research [20] shows that tampering of a physical device can be carried out through probing, material removal techniques, contactless radiation imprinting, etc. These attacks exploit physical and chemical properties to gain illegitimate access to a system.

Physical attacks [21] on systems can result in data theft, counterfeiting and cloning. Captured data is reassigned to a cloned device and then a verifier is convinced of the device legitimacy. A concern with cloned devices is that often their use can go unnoticed. Cloning and counterfeiting can be defeated through strong encryption and by enforcing restricted access to decryption keys [22].

2.2.2 Attacks on Communications

Communication based attacks allow an adversary to gain network access as a user or host, following which privileges are obtained leading to authentication and authorization abuse. Once access is obtained, an attacker may attempt to capture the cryptographic keys of the system.

IP spoofing [23][24] is a common attack in networks where an attacker forges IP addresses thus leading to falsified IP packets. Done correctly, an attacker can capture, reroute, modify or delete data in the network. IP spoofing is particularly damaging because it is an online camouflage attack which is often difficult to detect. A recent attack in the USA called the Dyn Cyber Attack resulted in internet outages at an unprecedented scale [25]. Webservers of several high profile social media sites, news agencies etc. were compromised. The attack was carried out by using the Mirai malware [26]. It is estimated that to carry out the attack about 100,000 malicious IoT devices were used [27]. The malware functions by identifying vulnerable IoT devices that are using the factory default username and password. Once a device is captured it is then used as a bot to inundate a remote server with large amounts of data to create Distributed Denial of Service attack (DDoS) [28][29]. The amount of data was so large that many websites reported incoming data of upto 1 Tbps.

Perhaps the most common form of attack on communication systems is eavesdropping. Many wearable devices transmit data wirelessly which makes them prone to eavesdropping. Eavesdropping can be defeated by ensuring that the data is encrypted when it leaves a system.

2.2.3 Attacks on Cryptosystems

Cryptographic algorithms often base security on the secrecy of keys. Adversaries can attempt to compromise a system by capturing its cryptographic keys. What sets cryptographic key theft apart from other forms of attacks on computer systems is the fact that when keys are duplicated there may be no evidence of the unlawful activity [30]. When data is duplicated there is often no trace of the activity taking place. Similarly, when a cryptographic key is duplicated there is no evidence that the key was duplicated. Further, when a stolen cryptographic key is used in an unlawful way then its use often goes undetected.

Research shows that cryptographic keys can be captured through a diverse range of attacks [9][10][31] like brute force, cold boot attacks, malware etc. As there are multiple methods of attack therefore key theft deterrence can be a complex task. Attackers attempt to exploit weaknesses or design flaws in a system to capture cryptographic keys. Given below are some possible attacks which can lead to key theft.

- An attacker may attempt to defeat a cryptosystem by using brute force, dictionary based attacks, rainbow table attacks, man in the middle, etc. Appropriate steps like increasing key size, incorporating salts, not using obsolete algorithms can prevent these type of attacks.
- An adversary may provide a malign key generation software (malware) so that the keys can be communicated to him. Detecting this attack is difficult because often the user is not aware of the presence of malware and because it may not be possible to identify a bad code in a program. There are other variants [32] of this attack which can have an adverse impact on a cryptosystem.
- It is possible for attackers to use someone else's public key and claim that it belongs to them. Certification authorities need evidence to show that

the key is not being used as a forged identity. Certification authorities have in the past mistakenly issued certificates to forgers owing to which there is a growing certificate revocation list in web browsers.

- Cryptographic algorithms are often founded on algorithmic intractability like being based on large prime numbers, factorability etc. If the keys generation algorithm is weak or poorly designed then attacking the keys could be easier for an adversary. It is important that the keys are generated by a trusted authority. For example a key can be generated by an adversary impersonating as a trusted authority. Doing so the attacker would not only have knowledge of the keys but he may deliberately create bad keys which do not possess the correct properties.
- Attackers may employ psychological manipulation, persuasion [33] to obtain the keys from their owners. It is vital that the keys are kept secret from both insiders and outsiders. Social engineering is a powerful tool and can be used to compromise security at various levels.

By no means is this an exhaustive list of possible attacks on the keys of a cryptosystem. Readers should refer to [34][35] for a further discussion on possible attacks on cryptographic systems.

2.2.4 Attack Statistics

It has been seen that some devices in the IoT are being marketed with insufficient security provisions [7]. A reason for this is a lack of understanding of why an attacker would attack a device in the first place. It is a known fact that data and information is a commodity in the underground economy [18]. Attackers will go to any lengths to capture data so that it can be sold to prospective buyers. The effects of attacks on the various types of computation systems have been widely recognized and published [36][37]. According to a report [38] 100 million healthcare records were stolen or compromised in 2015. The report demonstrates that these health records contain a wealth of information like credit card data, email addresses, social security numbers and employment details just to name a few. This data is captured to commit fraud by stealing medical identities. An attack analysis shows that 15.5% of attacks were carried out by inadvertent actors. These insiders were either duped or lured into performing actions which can result in a security breach. Often an employee or subcontractor will give away information due to either incompetence or ill will. Given in figure 2.2 is a pie chart showing the breakdown of attacks on healthcare systems.

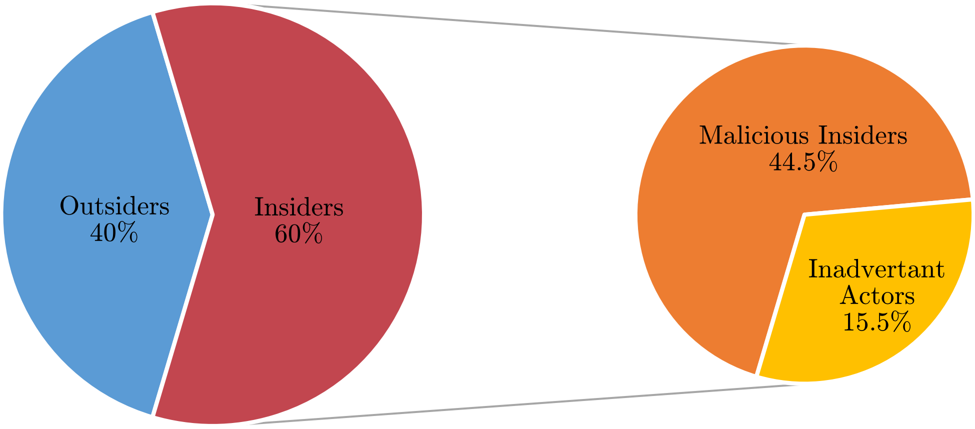

Figure 2.2. A breakdown of attacks on healthcare systems in 2015 [38]

Unauthorized access dominated the list of incident categories [38] with 45% while malicious code came second with 29%. Other forms of attacks have been identified but these attacks had significantly lower impact. From the incident categories it can be concluded that attackers attempt to capture valuable data remotely by exploiting weaknesses in the system.

2.3 ATTACKS ON WEARABLE TECHNOLOGY

In [39] the author has studied the wearable technology industry both technically and statistically. The author concludes that wearable technologies have three limitations which are barriers to their wide adoption i.e. battery life, chipset limitations, design concerns. Owing to these limitations wearable devices face many challenges among which data security and privacy is an important one.

IoT devices are finding their way into many different fields one of which is healthcare. Research [40] shows that the IoT will transform the way the healthcare industry works. In [41] the authors present a study on how wearable devices can improve working of challenging environments like hospital wards. They present a case study in which they conclude that wearable devices would enhance the level of usability and context awareness. The authors have identified four security challenges facing wearable devices i.e. confidentiality, authentication, hostile environment and device network security.

Wearable devices incorporate practical features and function by using the latest technologies and trends. Wearable devices are being used and experimented with to facilitate the wearer through various ways [42]. A recent research [43] on fall detection through inertial sensing has been studied by Kumar et al. The authors design an assisted living wearable device embedded with a tri-axial accelerometer and tri-axial gyroscope. By using these two sensors the wearable device can sense linear acceleration and angular velocity to detect falls in the elderly or disabled. The wearable device facilitates communication of data by using Bluetooth. Being a health monitoring sensor the system continuously senses motion related variables, but the system lacks any form of security implementation.

There are multiple ways of authenticating the wearer of a device one of which is through gait recognition. Authenticating wearable devices using gait recognition is a concept which has been explored [44] [45] in much detail. Chauhan et al. in their paper [46] design a security scheme for the optical wearable device Google Glass. The authors present an unobtrusive security scheme which uses multiple user gestures to establish user authenticity. Although the concept is interesting it has a weakness that it requires user intervention for authenticity. Another weakness of the proposed scheme is that the user must possess prior experience with the Google Glass for improved accuracy.

Authentication can also be carried out by sensing bioelectrical body signals. Researchers have developed a scheme [47] which can detect the wearer of a wearable device by using the bioelectrical impedance signal. The research shows that it is possible to use a wrist wearable health sensor called the Shimmer sensor [48] to uniquely identify a user by using their physiological signals. The proposed scheme possesses a 98% successful authentication rate but the scheme does not offer other basic security services like integrity and confidentiality. The provision of authenticity alone is a false promise of security and hence the work needs extension.

A recent research [49] shows that even widely marketed wearable devices can possess poor security provisions which makes attacking them an effortless task. The paper studies the Fitbit tracker that has 96KB RAM and is embedded with an accelerometer sensor, altimeter sensor. The paper studies the security of the Fitbit tracker and shows that it is possible to attack the wearable device by exploiting weaknesses in the system. The authors reverse engineer the Fitbit and observe that it lacks security provisions. For instance the tracker transmits user credentials in plain text. Besides this any HTTP data processing that takes place is also in plaintext. The authors also demonstrate that counterfeit data can be generated and injected into the tracker by attaching it to moving objects like the wheel of a car.

Devices in the IoT are not just limited to wearables. Devices of many forms are available which can be installed in various settings and accessed remotely. In [7] the author demonstrates practically how to attack various systems in the IoT. The author demonstrates how to attack common IoT enabled systems like home lighting, electronic door locks, baby monitors, smart televisions, and smart vehicles. The study on various IoT enabled systems shows that security weaknesses are not just limited to low priced systems. The author has taken a 416 horse power Tesla S P85+ electric car and demonstrates how it can be stolen through multiple methods of attack like password theft, API adaptation and network based exploits. Similarly, the author demonstrates that often weaknesses are found in systems because of poor design. For instance a Samsung Smart television allows users to upgrade its firmware. Studying [7] the firmware shows that the firmware is encrypted using a flawed implementation of XOR cipher [50]. In the implementation a key much smaller than the plaintext is used which means that a large portion of the plaintext is never encrypted. Weaknesses in IoT capable devices shows that multiparty systems need to be redesigned for improved security, privacy and safety.

2.4 PHYSICALLY UNCLONABLE FUNCTIONS

It is a known fact that no two silicon chips are created alike [51]. Even if the manufacturing, design, materials are the same the resulting chips vary from each other considerably. A reason for this variability in the chips is uncontrollable and unavoidable variation at the molecular level. These variations are employed for creating a one way function called Physically Unclonable Functions (PUF). A PUF is a function based on a physical property and holds the quality that it is unclonable [52]. When a PUF is queried with a challenge x the function provides a secret response y such that the response is based on the unique characteristics of a device. Hence a PUF produces an unpredictable output which is based on the underlying physical properties of the device. A PUF exploits the variability in chip manufacturing to create an unpredictable output that is characteristic of the particular device [53]. Figure 2.3 gives a generic depiction of a PUF.

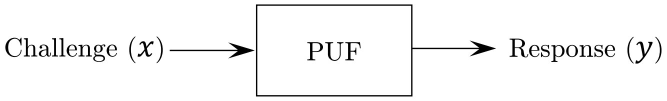

Figure 2.3. A generic PUF with the challenge x and associated response y

PUF's possess qualities like robustness, unclonability, unpredictability and tamper evident design which makes them an attractive technology for use in cryptography. Given below is a description of PUF qualities in a security system:

- Robustness – when queried with a single challenge, the PUF must produce similar responses with a high probability.
- Unclonability – it should be infeasible for an adversary to produce two PUF's that produce a single response to a single challenge.
- Unpredictability – it should be infeasible for an adversary to predict the response to a challenge even if the adversary has previously queried the system multiple times.
- Tamper evident – if an adversary attempts to tamper with a PUF then this should change the challenge-response behaviour.

Owing to their unique properties and design, PUF's have become very attractive for use in a variety of security related applications [54][55]. The security of cryptographic schemes is based on mathematical problems that are now under attack due to the creation of new computing architectures and algorithms. Research studies [56][57][58] have shown that PUF's can be used for hardware entangled cryptography, authentication, IC-identification, anti-counterfeiting, and random number generation. Since a PUF is an unmodifiable function therefore it can be used to prevent overclocking and also detect whether the binding between hardware and software is in conformance with a manufacturers recommendations [59]. The use of a PUF for key generation offers the greatest flexibility because a PUF is non-volatile yet at the same time it does not have the problems associated with data storage and data theft. Thus PUF's can provide intrinsic key storage that is hardware associated [60].

Research is underway to identify unique characteristics that can be used for creating strong PUF's. Early research [61] showed that optical PUF's can be created by detecting the splatter pattern from a stationary scattering medium placed in the path of a laser. In this application the input is the placement of the laser beam in the x-y plane while output is the associated splatter pattern. This research practically demonstrated the establishment of PUF but had limited useful applications. Research [62] has shown that it is possible to use RFID as a PUF. The authors demonstrate that a 64 bit input challenge can be used to create a unique response. The PUF is designed with a scrambling circuit to help prevent learning based attacks on the PUF output. Experiments on manufacturing variability in logic gates shows that a delay in the circuit gates can be used as a PUF [57]. The authors have observed that logic gates are influenced by factors like supply voltage and operational temperature. These parameters are prone to change which is why the PUF is classed as weak. Experiments [63][64] on using Static Random Access Memory (SRAM) cells as a PUF have shown that each cell has a start-up state of either zero or one. This state is unpredictable which makes it very suitable for generating a unique device fingerprint also known as SRAM PUF [65].

The use of PUF has been studied in cryptography. Research shows that a PUF can be used for authentication [66], secure key storage [67], key generation, key zeroization [68] etc. It is recognized that using PUF to support cryptographic functionalities can provide increased flexibility, security, reliability while reducing cost and storage needs [67]. Researchers [69] have studied the design and implementation of a PUF based cryptographic key generator. The key generator uses a modular design that is based on a ring oscillator. Tests on the PUF have shown that it has 99% entropy coupled with a low overhead. The authors have shown that the designed tool is highly adaptable but they have not shown technically if the PUF can be trusted as a basis for a cryptographic key. The authors conclude their work stating that it is information-theoretically security.

The viability of PUF is recognised in computationally advanced devices owing to the availability of a wide range of measurable features. Research [70] has shown that PUF's can be used in low end embedded devices for eliminating anti-counterfeiting and software manipulation. The research aims to secure devices that are commonly available, lack resources and are low priced. The research uses an SRAM PUF for creating a secret with full entropy. The authors have shown that their software implementation uses hash functions and on average these will add a 63% overhead to the existing software. This is relatively high considering the fact that many of the devices may lack sufficient resources. A positive point of the work is that the performance overhead is not too excessive at 10%.

2.5 IDENTIFICATION THROUGH FINGERPRINTING

Biometric fingerprinting [71][72] is the process of identifying individuals by using their fingerprints. Biometric fingerprints uniquely identify an individual because of unique placement of lines and ridges on the fingers.

Device fingerprinting follows a similar concept by generating an identification for a device using features that help distinguish it from other devices. Motivation for device fingerprints stems from the broad-spectrum importance of biometric fingerprinting. A recent yet broad definition of a device fingerprint has been given in RFC 6973 [73]. The RFC offers guidance on privacy consideration for internet protocols and defines a fingerprint as follows:

Definition 2.1. The *fingerprint* of a device is defined as a set of information elements that identify a device or application instance [73].

Although the definition is fairly broad it can be concluded that the purpose of a fingerprint is to identify a device (with a sufficiently high probability). Thus to identify a device the choice of identification elements plays an important role and should not be limited to only the hardware environment. All computation devices possess subtle but measurable variations which can be obtained to create fingerprints. Commonly referred to as device fingerprints or hardware fingerprint, the purpose of these fingerprints is to identify an individual device, system or a user with a high precision. There are many methods of uniquely identifying a device. A common method of device identification in a web based environment is achieved through the use of HTTP cookies [74]. When used constructively, a cookie allows a web server to store small piece of information on a client system. This file is then sent back to the server when subsequent connections are established. The purpose of cookies is to track the user activities and browsing habits to customize browsing sessions. Destructive use of cookies can undermine user privacy and allow attackers to tailor exploits according to the installed browser, plugins, applications and operating system [75].

Seminal work on web browser fingerprinting [76] shows that fingerprints can be created using configurations found in a browser. The author demonstrates that a fingerprint can be created using unique browser features like fonts, screen resolution, timezone, browser plugins, canvas, WebGL, etc. to identify a browser with fairly high precision. When a user switches browsers the fingerprint also changes, which may seem like a limitation but one must acknowledge that most users tend to have a single favourite browser therefore the browser fingerprint can be used as a device fingerprint. The outcomes of the research have design implication both for privacy and technical design.

A recent research [77] demonstrated that it is possible to fingerprint mobile devices using personalized configurations found in the device. The authors have identified 29 unique features which can identify a device with 97% accuracy. The authors use a unique set of features like WiFi SSID, device model, device name, network carrier name, twitter account name, songs list, etc. to create a device fingerprint.

A simple amalgamation of device features does not guarantee a useful fingerprint. To be effective, machine fingerprints must possess two qualities i.e. diversity and stability. These two qualities form guidelines for the formation of a fingerprint that is exclusive and inimitable. The diversity of a fingerprint can be studied by measuring its entropy. While stability of a fingerprint can be verified through rigorous device testing to prove resilience to change. Given below is a definition of diversity and stability.

Definition 2.2. The *diversity* of a machine fingerprint is the quality that no two devices have the same fingerprint. The more features used for creating a device fingerprint, the more likely it is to obtain a distinguishing fingerprint.

Definition 2.3. The *stability* of a machine fingerprint is the quality that the fingerprint remains constant over time. The more features used for creating a device fingerprint, the less likely it is for the fingerprint to remain stable.

Capability, complexity and resources of the target system dictates whether the features of a device will simply be collected or extracted through an intricate methodology. Given below is a breakdown of feature extraction and classification methods.

2.5.1 Client/ Server Models for Feature Extraction

Conventional device fingerprinting techniques are mostly web based and follow one of two models either client based or server based [78]. When using the client based method; the device features are extracted by installing software on the client device. The problem with the client based method is that installing the software requires user permission which may not always be possible since many users and organizations prohibit software installations. Installing the software is prone with its own dangers since the software may be a malware concealed in a seemingly meaningful application.

The server based method does not require a software installation because device identifications are generated by gathering device characteristics that are readily available and may not require user permission. The problem with this technique is that the device identifications are assembled using relatively simpler features which do not ensure diversity and entropy. Features used in the server method can be extracted and reproduced by attackers thus aiding spoofing. An example of server based device identification is the use of browser cookies which use stored information and credentials to identify users and sessions [79].

2.5.2 Intrusive Feature Extraction

Based on the level of intrusiveness, there are two methods [80][81] of feature extraction i.e. active fingerprinting and passive fingerprinting. Active fingerprinting actively queries the system for information required for establishing the fingerprint. In active fingerprinting the stimulus may be applied as an intrusive method of querying the system.

Passive fingerprinting establishes device features through less intrusive methods like monitoring a communication link. Most communication based fingerprinting methods are passive in nature and establish a device fingerprint by using network and packet information [82][83].

Passive and active fingerprinting offer benefits in different areas of application. Active fingerprinting offers more accuracy [81] as it has the ability to examine a wider range of behaviours which cannot be obtained using passive methods of fingerprinting.

2.5.3 Feature Classification Methodologies

Traditionally, device fingerprints are generated using features such as MAC addresses, serial numbers, OS fingerprint, cookies etc. The complexity of the selected features influences the unpredictability of the generated device fingerprint. Recent research [84][85] explores for the purpose of identification implicit features and proves that it is possible to identify a device using internal features. Broadly device features are placed into two categories i.e. explicit and implicit.

Explicit – those fingerprints which are established using well-defined and standardised features outlined by the manufacturer. These features can include serial numbers, MAC addresses, firmware versions and clock frequencies. These features are simpler to extract owing to which they can be easily predicted and spoofed. A challenge with these features is that they often appear on the exterior of the device (MAC address, IMEI, serials etc.) which makes compromising the resulting fingerprint an effortless task. Even if these features do not appear on the exterior of a device they can be extracted using a combination of network monitoring and analysis tools.

Implicit – those fingerprints which are established using less obvious features. These features may be a result of inconsistencies in the device fabrication processes. For example the clock skew varies in every device even though the clock frequency remains the same for a particular device and model. Implicit features are unique and low level features which are not easy to predict thus making spoofing a difficult task.

2.6 INTEGRATED CIRCUIT METRIC

According to the Kerckhoff's [86] principle, "the security of a system should lie in keeping the key secret and not the algorithm". Whenever a cryptographic algorithm is designed it is widely published so that it can be studied for conformance to the highest levels of security. If the algorithm is published then the only crucial element keeping the system secure is the cryptographic key. A cryptographic key is selected from a key space such that the key space is large enough to prevent brute force attacks while every key has the same possibility of being selected. Ensuring that the keys are generated at random simply certifies that the keys are unique but does not defeat the possibility of key theft. Also an increase in the key entropy makes it difficult for adversaries to brute force but does not eliminate the possibility of key theft. This implies that a stored cryptographic key is an Achilles heel in conventional security systems. Traditionally, attackers will breach a system and attempt to capture the secret key. Once the key is captured then decryption and other cryptographic operations are trivial tasks which can be carried out without much effort. Besides this attackers can attempt to gain illegitimate access to data and network through a wide range of attacks as discussed earlier in this chapter.

Two methods of ensuring device security have been identified in the literature [87]. The first method of security attempts to authenticate the user wearing the device while the second method aims to identify the device and thus secure it. These are two different paradigms where the first ensures only the security of the wearer. The second method secures the device which leads to the security of its wearer. As cryptographic keys are stored on a device therefore focusing on just the device or the user can result in a flawed security implementation.

The Integrated Circuit Metric (ICMetric) technology [88][89] has been conceived as an alternative method to stored keys and as a basis for a range of cryptographic services. The unique concept and design of the ICMetric technology does not limited its use as an alternative method to stored keys. Research [88][89] on the ICMetric technology shows that it is both possible and recommended to use the features of a device to generate an ICMetric which can then be used for the provision of cryptographic services in a system. The ICMetric technology deters key theft by entirely eliminating the need for stored cryptographic keys. By using the ICMetric technology there is no need to store the keys or any associated templates because the ICMetric and keys are generated when required and discarded thereafter. Doing so discourages attackers since there is no cryptographic key present on the system. The ICMetric technology bears close resemblance to biometric systems, as these systems use identifiable features to identify different persons. Similarly the ICMetric technology proposes using device features to identify every device uniquely. The ICMetric technology achieves this without the need for stored templates or associated data. This quality means that the ICMetric technology can be used for preventing key theft, impersonation and spoofing based attacks on computation systems.

The ICMetric technology processes unique measurable properties and features from a device and provides as output an identification formally called the ICMetric. The creation of an ICMetric is a complex process primarily because the ICMetric is based on both explicit and implicit features. The security of an ICMetric based system relies heavily on the features employed for ICMetric generation. For instance even though the MAC address uniquely identifies a device, it is not a strong candidate because it can be easily extracted using a network surveillance tool like Wireshark. Owing to this, the ICMetric is generated using a range of low level features of a device. Using low level features has the advantage that these features cannot be easily predicted or replicated by an adversary. Previous studies [90][91][92] have identified hardware features which can be used for ICMetric generation. Experiments show that the Program Counter (PC) and Cycles Per Instructions (CPI) can used to generate an ICMetric. The use of these features for ICMetric generation has not been investigated on smart devices.

As the ICMetric is a form of device fingerprint that will enable various security applications therefore it must possess certain qualities as follows:

- Unique – the ICMetric of a device must be unique.
- Reproducible – the ICMetric of a device must be a reproducible i.e. the same ICMetric must be generated every time it is required.
- Deterministic – the features used to generate an ICMetric must have a deterministic range. This ensures that the features produce consistent readings resulting in a stable and reproducible ICMetric.
- Self-producing – a device should generate an ICMetric without the need for special instrumentation or user intervention.
- Non-disruptive – the ICMetric should be generated without disrupting the regular functioning of a device.
- Non communicability – to protect from attacks the ICMetric cannot be communicated even to trusted entities.

Not all feature of a device are suitable for creating an ICMetric. The features should possess certain qualities that make them suitable for the purpose. A multivalued feature must possess the following qualities to qualify as a candidate for ICMetric generation.

- Feature values can map onto a unimodal distribution i.e. feature values must not be erratic in nature.
- Features can possess a Gaussian distribution.

Multimodal distributions possess two or more pronounced peaks in response to a single stimulus. Multimodal distributions present much greater challenges owing to overlapping observations which often implies that a data sample lacks homogeneity. Processing this dataset requires complex algorithms [88] which can reduce the practicality of an ICMetric based system.

As the ICMetric of a device is unique, one may be tempted to use the ICMetric as a cryptographic key. The ICMetric of a device cannot be used as a cryptographic key because it lacks size and entropy making it susceptible to attack [93]. The ICMetric technology is designed as an extra security layer that aims to change how cryptography has been implemented in computation devices. The purpose of designing the ICMetric technology as a layer is to provide seamless connectivity to the existing security infrastructure, technologies and algorithms. Thus the ICMetric technology aims to enhance security with minimum impact on existing system operations. An extended ICMetric based IoT communication suite is given in figure 2.4. The ICMetric layer connects with the perceptual layer so that device features can be obtained from physical world. The ICMetric layer enables feature extraction, ICMetric generation and ICMetric security.

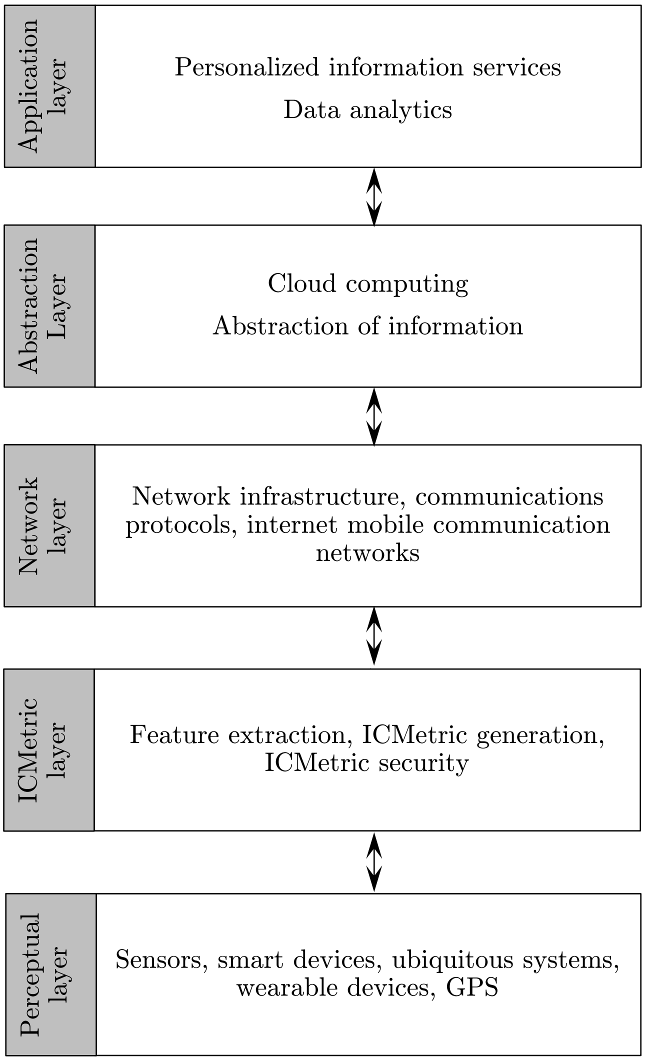
Application layer
Personalized information services
Data analytics
Abstraction Layer
Cloud computing
Abstraction of information
Network layer
Network infrastructure, communications protocols, internet mobile communication networks
ICMetric layer
Feature extraction, ICMetric generation, ICMetric security
Perceptual layer
Sensors, smart devices, ubiquitous systems, wearable devices, GPS

Figure 2.4. IoT communication suite with an ICMetric layer

2.6.1 ICMetric Generation

Generation of an ICMetric begins once the features of a device have been extracted. Individual device features are extracted and then feature sets are established to process statistically. The ICMetric generation is a two-step process composed of a calibration phase and an operational phase. These phases are applied only when required following which the ICMetric and any associated data are discarded. The ICMetric and associated feature data is never communicated during any phase of generation or use.

Step 1 - Calibration Phase

- Suitable features are selected to obtain readings for a device. The individual readings are used to define feature sets to be used for frequency analysis.
- Compute frequency distributions for each feature set.
- Create histograms using the frequency distribution.
- Compute statistical credentials from individual histograms.
- Combine credentials to generate device ICMetric.

Step 2 - Operation Phase

- Extract feature values required for ICMetric generation.
- Generate the histograms by first computing the frequency distributions for the feature sets.
- Apply cryptographic key generation scheme for the provision of security services.

2.6.2 Combining Features

Individual feature values/ credentials need to be combined so that a final ICMetric can be computed. Two techniques have been identified [94][95] which can be used for combining feature values i.e. feature addition and feature concatenation.

2.6.2.1 Feature Combination through Addition

In feature combination through addition the ICMetric is generated by adding the individual feature values. A benefit of this technique is that the resulting ICMetric is highly diverse because adding a feature introduces variance but has low impact on the size of the number. If icm_d is the ICMetric of a device and F is a feature then the ICMetric using n individual features can be represented as follows:

$$icm_d = \sum_{i=1}^{n} F_i \qquad (1)$$

2.6.2.2 Feature Combination through Concatenation

An alternative to the feature addition technique is the feature concatenation technique. In this technique the individual feature values are combined using the concatenation operation. The resulting ICMetric lacks diversity but has a longer length [94] as incorporation of each feature increases the overall length. If $\|$ is the concatenation operation then the device ICMetric icm_d using n individual features can be represented as:

$$icm_d = F_1 \parallel F_2 \parallel \cdots \parallel F_n \qquad (2)$$

2.6.3 Comparing ICMetric and PUF

The ICMetric technology and PUF share a similar concept but in reality they are different. The ICMetric technology uses measurable hardware and software features for creating a device identification. The

ICMetric technology does not always rely on a challenge and its associated response. This implies that the ICMetric technology can be used to find correlations between various system elements which may not even take a challenge as an input. Further, the ICMetric technology extracts features and then processes them to produce a single device identity. As the ICMetric of a device can be used to generate keys therefore it can be said that the technology is a key theft deterrent and also provides authenticated communications using asymmetric keys (private key encryption coupled with public key decryption).

Commonly PUF's are hardware oriented and use a challenge-response setup to establish legitimacy. The challenge-response naturally places a limit on the number of features which can be used as suitable PUF's for securing computation systems. This limit comes from the fact that not all system elements are designed for querying. An example of this is single line identification chips [96]. Compared to an ICMetric multiple responses from individual PUF's are combined using an XOR arbiter PUF [97]. Another factor that is worth mentioning here is that if a PUF is based on an analogue system then it is highly susceptible to noise.

2.7 SUMMARY

This chapter has established the importance of security in the group environment. The chapter takes the example of the emerging IoT as a collection of smart objects. As the IoT is composed of multiple interconnected devices that function ubiquitously in a heterogeneous environment therefore special emphasis has been placed on the security of wearable devices. To highlight the problem associated with security of groups it has been proven that when devices communicate in a group environment they can be an easy target for adversaries.

Convenient access to computation power means that adversaries are stronger than before and they can attack devices through multiple methods. This chapter demonstrates with the help of research studies that some devices in the IoT possess insufficient security provision. It has also been shown with examples that those devices which offer security, possess flaws which can be exploited by adversaries. There are numerous ways through which a system can be attacked, therefore this chapter establishes that adversaries will attempt to exploit physical, communication or cryptosystem weaknesses to gain illegitimate access. Often the goal of adversaries is to attack a system so that the cryptographic keys can be captured. Effort to increase the key entropy is not a practical approach since attackers can capture the key by exploiting network or physical weaknesses in the system.

This chapter explores device fingerprints and physically unclonable functions as a prequel to hardware entangled cryptography. The chapter then presents an in-depth study on the novel ICMetric technology. The ICMetric technology has been studied as a key theft deterrent technology and as a basis for a range of cryptographic services. The technology proposes using unique features of a device to create an identification called an ICMetric. The ICMetric technology processes unique reproducible explicit and implicit features so that the resulting ICMetric is truly unique. The two phases of ICMetric generation have also been presented for processing a range of possible features which are explored in the upcoming chapter.

CHAPTER 3

ICMETRIC - A FEATURE STUDY

Recently, there has been much interest in the area of device fingerprinting. The area of research is in its experimental incarnation, and its attraction stems from many sources. Among these is the hope that machine fingerprinting will provide a much needed identification to the many devices on the internet. Additionally there are many benefits for telecommunications, cryptography and forensic sciences.

A difficulty associated with device fingerprinting complimented cryptography is the lack of formally specified algorithms, models and protocols that guarantee high levels of security. There are numerous features in a device which can be used for identification, but these features lack complexity which is why they are likely to be captured by adversaries.

The ICMetric technology is not just a device fingerprinting technology. The purpose of this novel technology is to create a unique device identification which can be used for a wide range of cryptographic services like authentication, key generation, confidentiality and integrity. Hence by using the ICMetric technology a device identification is created which is used for the provision of cryptographic services. The security of an ICMetric based system relies on finding features suitable for generating a device ICMetric. If this is done incorrectly the resulting system gives a false guarantee of security. This chapter presents a study on unique explicit and implicit features which can be used to generate a device ICMetric. Many modern devices utilize embedded MEMS sensor for the provision of a wide range of services. MEMS sensors like the accelerometer, gyroscope and strain gauge possess a bias which is noticeable in the readings obtained from the sensor. This chapter presents a study on the bias found in MEMS sensors and shows that the bias can be used for ICMetric generation.

The ICMetric of individual devices can be used to create a group identity. This identity called the group ICMetric is generated using the Shamir secret sharing scheme. This chapter also demonstrates how the group ICMetric is generated while preserving ICMetric secrecy of the individual devices.

3.1 MICRO ELECTRICAL MECHANICAL SYSTEMS

Micro Electrical Mechanical Systems (MEMS) is the name given to miniaturization of sensors that are designed using a combination of mechanical and electrical components [98]. The physical size of a MEMS component can range from less than one micron to several millimetres. What sets the MEMS based sensors apart from other sensors is the fact that they use a combination of electrical and mechanical components to sense physical characteristics. The sensors sense and convert mechanical, thermal, magnetic, optical, chemical phenomena into digital readings by using specialized electrical components. A typical MEMS system is composed of four components i.e. Micro sensors, micro actuators, microelectronics and micro structures.

The accelerometer and the gyroscope are the most widely used MEMS sensors and have a wide range of applications [99]. These sensors are being embedded into smartphones, laptops and vehicles. In smartphones the accelerometer and gyroscope are used to enable motion recognition [100].

Smartphones use accelerometer and the gyroscope as a source of additional information to enable rotation and tilt detection. In laptops the accelerometer and gyroscope is used to sense freefall/ movement so that the hard drive head can be paused thus preventing damage to the head or the surface of the disk. In vehicles the accelerometer is used to activate air bags by detecting a spike in acceleration which is an indicator of a collision. A gyroscope is embedded in vehicles to enable electronic stability control features so that roll overs can be prevented [101].

This chapter studies the possibility of generating an ICMetric by using low level features from MEMS sensors. Experiments on unique features for the generation of an ICMetric is based on a wearable health device called the Shimmer sensor [48]. The sensor is a rechargeable battery powered device embedded with an accelerometer, gyroscope, strain gauge, ECG and EMG sensor. The sensor has a sampling frequency from 5Hz to 50Hz and data is communicated via Bluetooth. The Shimmer sensor is supplied with two straps so that it can be worn either on a wrist or around the waist. Before delving into the experimental details it is important to first understand the internal working of the MEMS accelerometer, gyroscope and strain gauge sensors.

3.1.1 MEMS Accelerometer

The accelerometer [102] is a capacitance based displacement sensor which is composed of fixed plates placed in a spring mounted movable mass. Capacitive sensors sense physical input by a change in capacitance. The change in capacitance is so miniscule that it can only be read using specialized electronics. The sensor is composed of plates that are suspended in a movable mass. A voltage is applied across the plates so that a change in capacitance can be measured when the sensor is subjected to an external force. Figure 3.1 shows that a voltage has been applied to the plates suspended in the movable mass. When an acceleration is applied the movable mass moves which in turn causes a change in capacitance between the suspended plates and the movable mass.

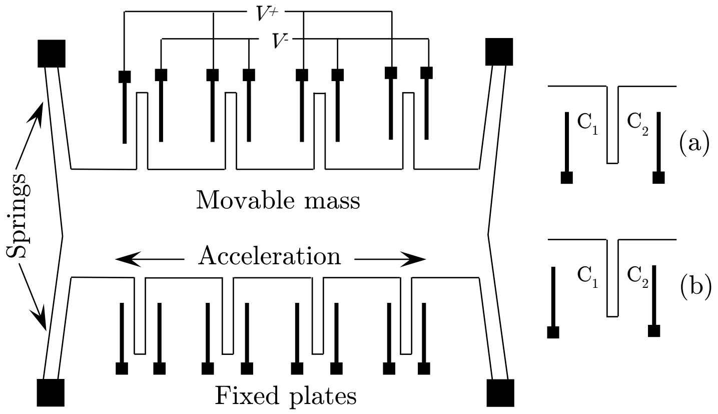

Figure 3.1. The working principle of a MEMS accelerometer (a) no acceleration results in the same capacitance on the fixed plates (b) acceleration causes change in capacitance between the fixed plates

Suppose a voltage V is applied to the fixed plates in the sensor. This voltage produces a capacitances C_1 and C_2 between the fixed plates and the movable mass. If the device is stationary or moving at a constant velocity then the two capacitances will be equal hence:

$$C_1 = C_2 \tag{3}$$

If the device experiences a change in velocity then both capacitances C_1 and C_2 will be different thus:

$$C_1 \neq C_2 \tag{4}$$

If a MEMS accelerometer possesses an imperfection it is reflected in the readings obtained from the sensor. Hence if an acceleration is applied to a sensors axes, then the sensed acceleration readings will differ from those being applied. The readings from a modern accelerometer represent the acceleration (m/sec^2)

along the three axes of motion. The Shimmer sensor follows this conventions and provides tri-axial accelerometer readings in a CSV file.

3.1.2 MEMS Gyroscope

The MEMS gyroscope is a sensor based on the Coriolis effect [103]. This effect is experienced by a body when it is subjected to velocity in a rotating frame of reference. Gyroscopes sense angular velocity with the help of drive arms that twist and rotate when they sense rotation. The drive arms are designed to be tall structures that resonate according to the sensed rotation. When a drive arm experiences axial rotation or lateral movement the drive arm is subjected to a Coriolis force and Coriolis acceleration. A gyroscope is designed so that the Coriolis force on the drive arm is proportional to the rotation speed in the particular frame of reference. Hence a gyroscope always measures the effect of a force experienced by the drive arm. Figure 3.2 shows the construction of a MEMS gyroscope and how the drive arm behaves when it experiences axial rotation and lateral movement.

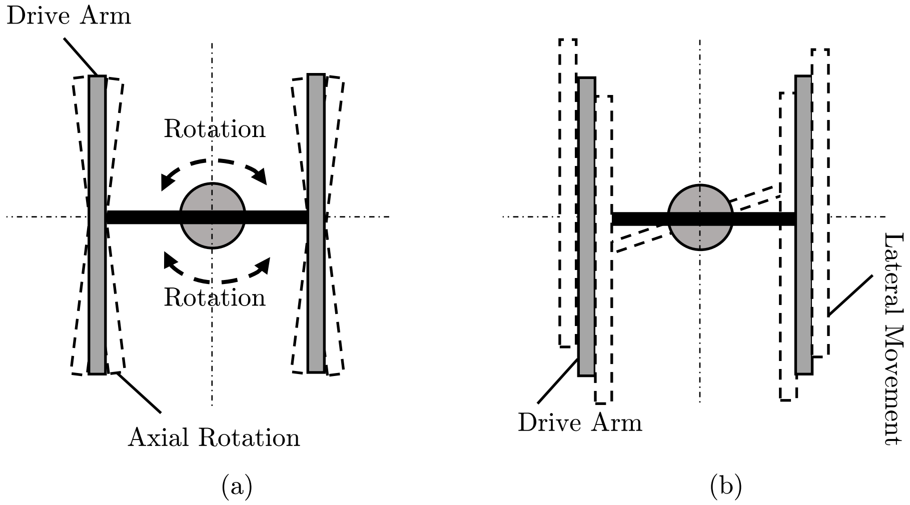

Figure 3.2. The working principle of a MEMS Gyroscope (a) drive arm movement with rotation (b) drive arm movement with lateral movement

Readings from a MEMS gyroscope can be used to detect the presence of a bias [104]. If a MEMS gyroscope possesses an imperfection it is reflected in the readings obtained from the sensor. Hence, if a rotation is applied to a sensor, then the sensed rotation readings will differ from those applied. The readings from a modern gyroscope represent the rotation (deg/sec) along the three axes of motion. The Shimmer sensor follows this conventions and provides tri-axial gyroscope readings in a CSV file.

3.1.3 MEMS Strain Gauge

The strain gauge is a unique sensor which is integrated to detect mechanical stresses and strains on a system. The sensor is commonly integrated into devices to enable decision support where mechanical stresses and strains are encountered. The strain gauge can be found in a variety of systems targeting health monitoring systems, to automotive, aerospace, wind turbine and other similar mechatronic systems [105].

The strain gauge sensor uses the relation between material properties and electrical conductance to measure the strain sensed by the sensor. If a conductive metal strip is stretched it will result in an increased end to end resistance. Conversely, if the metal strip is compressed it will result in a reduced end to end resistance. The stretching and compression must be reasonable so that the internal conductor does not permanently buckle or compress thus damaging the sensor altogether.

A typical MEMS strain gauge uses a thin metal foil in a strain sensitive pattern. When a strain is experienced by the sensor it will result in a change in the resistance thus exhibiting a change in the voltage. Figure 3.3 shows the construction of a MEMS strain gauge sensor.

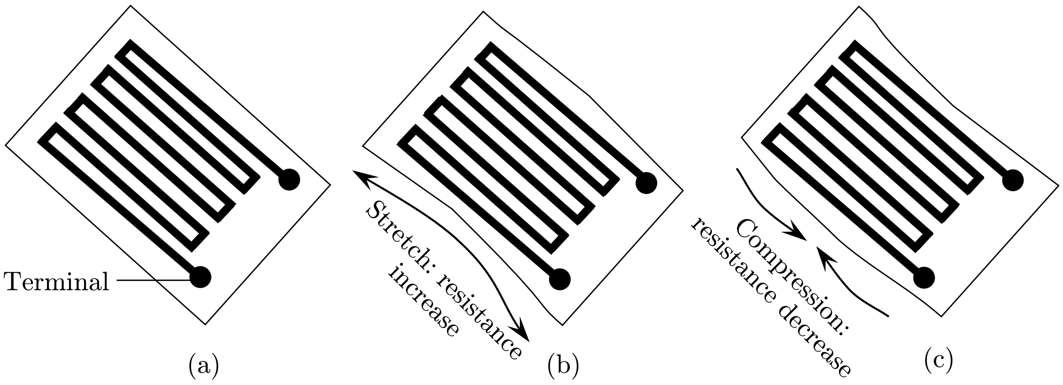
Terminal
(a)
Stretch: resistance increase
(b)
Compression: resistance decrease
(c)

Figure 3.3. The working principle of a MEMS strain gauge sensor (a) a sensor without stresses applied (b) stretching causes an increase in resistance (c) compression causes a decrease in resistance

Although the strain gauge sensor is used to detect mechanical stresses and strains, it can also be used in situations where a device has been attacked physically. If a hardware component embedded with a strain gauge sensor is attacked physically then the sensor can detect differences in the original and the residual stresses on the system. To detect if a sensor has been physically attacked a threshold must be defined to determine the acceptable operational strain range for the sensor. A typical strain gauge sensor will operate such that the sensed strain is on or around the median of the operational range. A strain gauge sensor provides readings in the form of high and low polarity voltage. The Shimmer sensor follows this convention and provides milliVolt readings in a CSV file. A strain gauge sensor also possesses imperfections which are reflected in the sensed and real strains experienced by the sensor.

3.2 HARDWARE IMPERFECTION ANALYSIS

Detecting the imperfections in a sensor through its readings requires comprehensive analysis of both the device and the embedded sensors. Not all sensors can be used for generating a hardware identification. There can be many reasons for this like some sensors do not possess adequate distinguishable features. Even if a sensor has distinguishable features an attacker maybe able to replicate the readings by placing another sensor in the vicinity of the target sensor so that very similar readings can be obtained [106]. Therefore a process is required that allows us to identify a sensor imperfection using implicit features which cannot be predicted or recreated by an attacker.

Hardware devices possess imperfections which are introduced when the hardware is being fabricated. When MEMS sensors are mounted onto the main board, stresses are applied which causes a permanent bias. Similarly, when a sensor is under operation its output accuracy is influenced by inconspicuous damages due to mishandling [107] and even the operational temperature [108]. Research [109] on MEMS reliability and failure methods has shown that there are ten individual types of mechanical influences that cause sensor bias while the electrical integrity is maintained. The bias in a sensor varies from sensor to sensor and is reflected in the readings obtained from the sensor. Calibrations attempt to compensate for the error in the readings by incorporating a linear value into the raw values obtained from the sensor. Recent research [12][110][111][112][113][114] on various sensors shows that it is possible to use the imperfections in a sensor to uniquely identify a device. Using sensor imperfections in conjunction with the ICMetric technology is a novel concept which this research explores with various types of MEMS sensors. Previous researches utilize the sensor bias for only identifying a device. The ICMetric technology is the first to utilize the sensor features and bias as a unique identification which is used as a basis for cryptographic services.

Computation devices possess many features which can be used to identify a device. The problem with using device features is that some features are too difficult to extract while other features may not uniquely identify a device. Generating an ICMetric for a device involves identifying features which can be reproduced only by the device. Below are some definitions necessary to understand how the implicit features of a device are collected from a MEMS sensor.

Definition 3.1. In statistics the *bias* is defined as the difference between the test result and the expected result. Hence the bias in a measuring instrument is the result of single or multiple systematic errors in the system [115].

Definition 3.2. Conditions in which independent test results are obtained with the same method on identical test items in the same laboratory by the same operator using the same equipment within a short interval of time. Hence the *repeatability* of a set of readings holds if precision is observed under the stated repeatability conditions [115].

Definition 3.3. Conditions where independent test results are obtained with the same method on identical test items in different laboratories with different operators using different equipment. Hence the *reproducibility* of a set of readings holds if precision is observed under the stated reproducibility conditions [115].

Different instances of the same device integrated with the same sensor will result in different readings being extracted even when the same stimulus is being provided [116]. Every sensor has a bias which can be verified by checking the output against a standard stimulus. Choice of a stimulus requires an investigation into whether the stimulus is easily created when required. Since the ICMetric should be generated without user intervention therefore a stimulus is required which does not require special apparatus or unusual actions by a user. For instance, the magnetometer bias can be used for generating an ICMetric but this particular sensor is greatly affected by the presence of electrical appliances like monitors, speakers, etc. The magnetometer is also affected by communication signals and flowing electric current. Therefore to use the magnetometer, an additional device is needed which would function like a Faraday cage. The purpose of the construct would be to isolate the magnetometer from external influences. Obviously, doing so would increase the complexity of the ICMetric generation and greatly reduce the practicality of the system. Table 3.1 shows components that possess imperfections but are not suitable for ICMetric generation due to technical reasons.

Table 3.1. Common components, their imperfections and why these imperfections are not suitable for creating an ICMetric

Component	Imperfection	Discard Reason
Magnetometer	Magnetic bias	Bias recreation and environmental influences on sensor
Clock skew	Idiosyncrasy in crystal oscillator	Low margin of error
Touchscreen	Touchscreen misalignment	Difficult to determine imperfection
Camera	Camera noise pattern	Limited inter device bias
GPS	Time skew between receivers	Not reproducible due to GPS latency
Flash Memory	Program disturbs	Need for a power cycle

The imperfections in a device are not limited to MEMS sensors. For instance the flash memory in a device has imperfections called program disturbs [117] which is the result of electrical stresses that are applied when programming other memory cells in the array. Program disturbs are not only rare occurrences but also require a device to power cycle (power off and power on) every time an identification is to be generated [118]. The need for a power cycle is counter intuitive and may not be suitable for healthcare devices or when a device has a long power cycle.

3.2.1 Experimental Details

To confirm whether a unique bias exists in the MEMS sensors, a sensor test bed is assembled which consists of five identical Shimmer sensors. Figure 3.4 shows the sensor testbed with one sensor plugged into a charging base station.

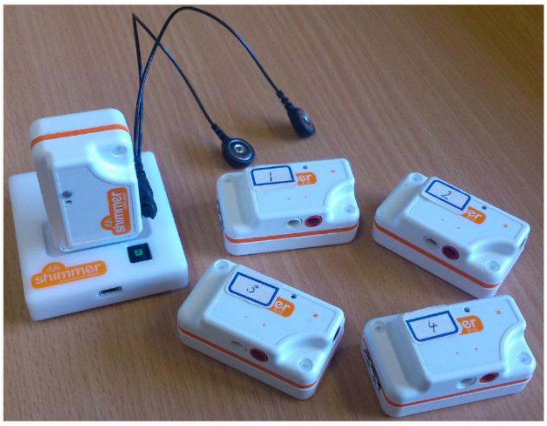

Figure 3.4. The sensor testbed composed of five identical Shimmer sensors

3.2.1.1 Methodology

To determine the bias in a sensor, 1500 individual calibrated readings per sensor axis are extracted. The readings are then used to create a frequency distribution which will result in a histogram that exhibits unimodal distribution [119]. A unimodal distribution is an asymmetric statistical distribution that possesses a single unique mode. Every normal distribution is a unimodal distribution, but every unimodal distribution is not a normal distribution.

The number of readings required is dependent on factors like sampling rate and accuracy of the sensor. Taking too little or too many readings is a concern since taking less number of readings may not adequately provide insight into the sensor behaviour. Whereas taking too many readings requires additional effort and will make insignificant events seem significant. Analysis shows that the number of readings influences the stability of the mean when compared to a target mean value. As the number of readings increases the stability of the statistic credentials is also attained. Figure 3.5 shows the effect number of readings has on the sample mean acceleration. Mean acceleration of the entire population is computed and used as a target mean or reference point. This target mean is used to determine how many readings are sufficient to attain statistical stability. Analysis shows that if the sampling is carried out under strict conditions then approximately 300 readings can be used to determine the bias in the sensor. A small sample size may not be adequate in situations where there is a risk of the sample getting contaminated.

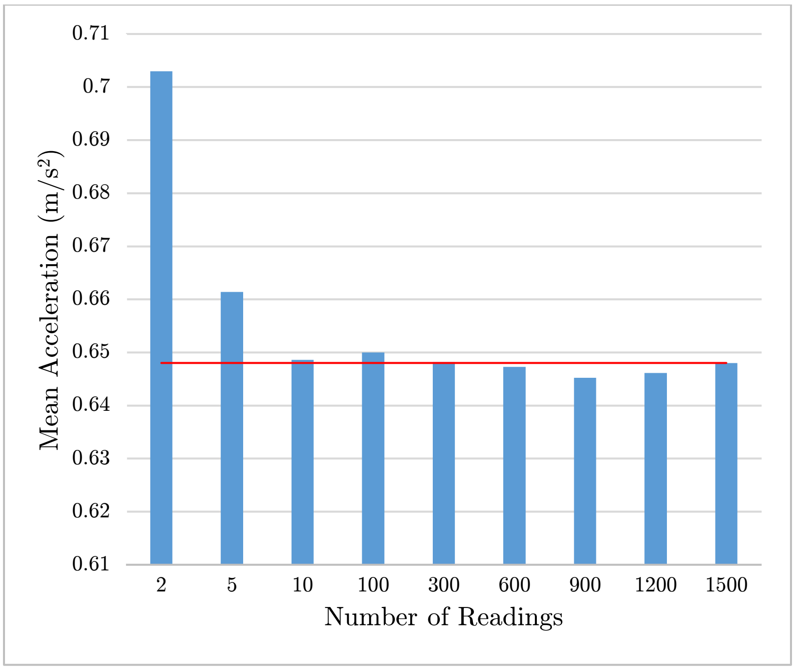

Figure 3.5. Number of readings versus the population mean acceleration

Sturges rule can be used to determine how many individual classes are required for the frequency distribution. Since the total number of sensor readings N is 1500 then, according to the Sturges rule the number of classes k is calculated as follows:

$$k = 1 + 3.3 \times (log_{10} N) \qquad (5)$$

$$k = 11.48$$

Based on the Sturges rule, 11 classes in the frequency distribution are created. After creating the histogram it is subjected to statistical analysis to prove that each sensor has a unique bias. The histograms follow a unimodal distribution which is then analysed using statistical measures like mean, standard deviation, confidence interval, kurtosis and skewness. To prove that the distributions are unimodal the readings are analysed using the Shapiro-Wilk normality test [120]. The test confirms that the sensor readings do not follow the normal curve. Statistical measures provide insight into the uniqueness of the sensor bias. The mean of a population shows where an average reading would lie in the population. The standard deviation indicates how widely dispersed the readings are compared to the population mean. Hence a low standard deviation indicates that the data points are close to the mean. The kurtosis shows how much the data set differs from a normal distribution. If the readings of a sensor follow a normal distribution closely then the kurtosis will be zero. A positive and negative kurtosis shows how much the peaks and tails differ from a normal distribution. Another strong indicator of how the readings are distributed is the skewness. If the skewness is negative then the tail of the distribution points to the left of the graph and vice versa. If the skewness is zero then this indicates that the distribution is symmetric i.e. the tail does not point to the left or right.

Statisticians often question where the mean would lie in a population based on a certain confidence level. This type of indication is of particular importance when a curve does not follow the normal distribution. The confidence interval determines the interval in which the population mean would lie based on a confidence level. There are three common confidence levels i.e. 90%, 95% and 99%. A higher confidence level increases the interval width so that it can be said with high precision that the resulting interval contains the population mean. Therefore, a trade-off needs to be made between being confident and widening the confidence interval [121]. To prove the uniqueness of the bias found in MEMS sensors the 95% confidence interval has been used. If $\bar{X}$ is the mean, N is the number of readings and σ is the standard deviation then the 95% confidence interval CI is given in equation 6. Here the numeric value 1.96 is the confidence coefficient for the 95% confidence interval.

$$CI = \bar{X} \pm 1.96 \times \frac{\sigma}{\sqrt{N}} \qquad (6)$$

To show that there is a significant difference between the axes of a sensor, analysis of variance (ANOVA) is conducted based on a single factor. The most important indicator of statistical significance in ANOVA is the p-value. The p-value of a sample forms the basis for the acceptance or rejection of the null hypothesis (there is no significant difference between populations). The p-value ranges between zero and one where low values indicate that there is a statistically significant difference existing in the values. If the probability equals one then this implies that statistically there is no significant difference between the set of readings [122][123]. A lower value approaching zero will indicate that the readings are significantly different from each other.

3.2.2 Accelerometer Bias Analysis

The Shimmer sensor is embedded with a tri-axial Freescale MMA7260Q [124] MEMS accelerometer. This accelerometer is a low cost micro machined sensor that has a sensitivity from ±1.5g to 6g.

To determine the bias in an accelerometer the Shimmer sensor is placed on a stable surface free from movements and vibrations. Precautions are taken that the sensor is not placed near or on an operating electronic appliance as this contaminates the resulting readings due to vibrations. The stimulus for assessing the accelerometer bias is subjecting it to 0 m/sec^2. This stimulus is easy to recreate by a user as there are many occasions when the sensor is left on a stable surface. An advantage of this stimulus is that a specialized device is not required for assessing the bias in the sensor. Under ideal conditions the readings from a sensor should be equal to the stimulus provided to the sensor. Experiments show that this is not the case and that each axis possesses a unique bias which is reflected in the readings. Experiments also confirm that the bias in the accelerometer is unique and reproducible provided the stimulus remains the same.

The accelerometer bias is a good implicit feature because it cannot be predicted for any particular sensor. Further testing confirms that there is sufficient statistical variances in the histograms obtained from the sensors. The statistical variances between different sensors makes the accelerometer bias an attractive implicit feature for ICMetric generation. Figures 3.6 - 3.14 show the calibrated acceleration histograms obtained from three identical Shimmer sensors bearing identifications 3B56, 3B79 and 3B81. From the graphs it is evident that each axis exhibits a different bias and that there is neither a similarity between the sensors nor a correlation between the individual axes.

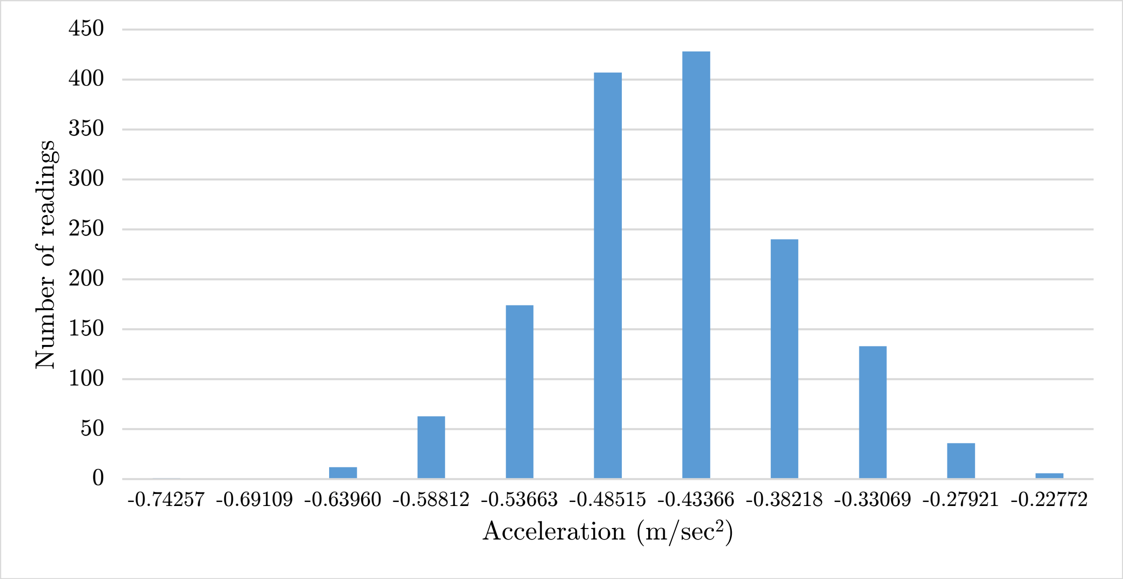

Figure 3.6. Calibrated x-axis acceleration histogram for sensor 3B56

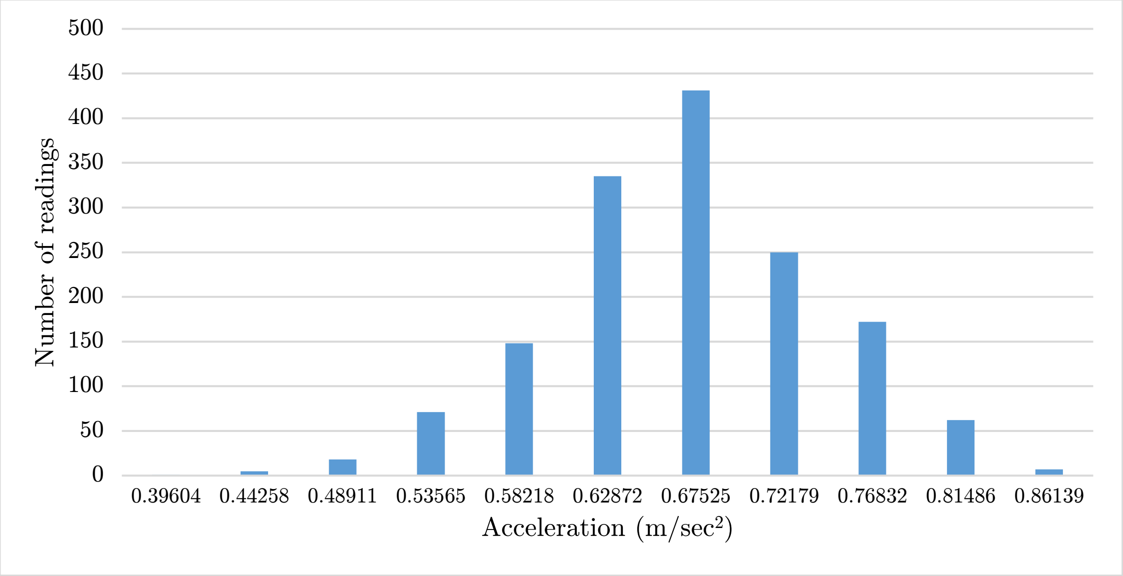

Figure 3.7. Calibrated y-axis acceleration histogram for sensor 3B56

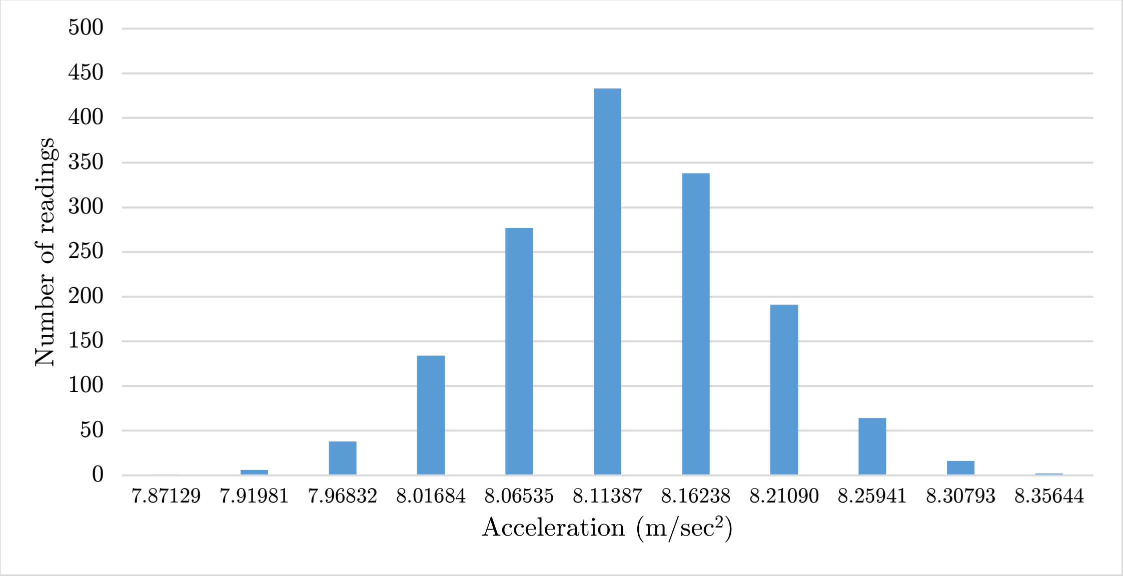

Figure 3.8. Calibrated z-axis acceleration histogram for sensor 3B56

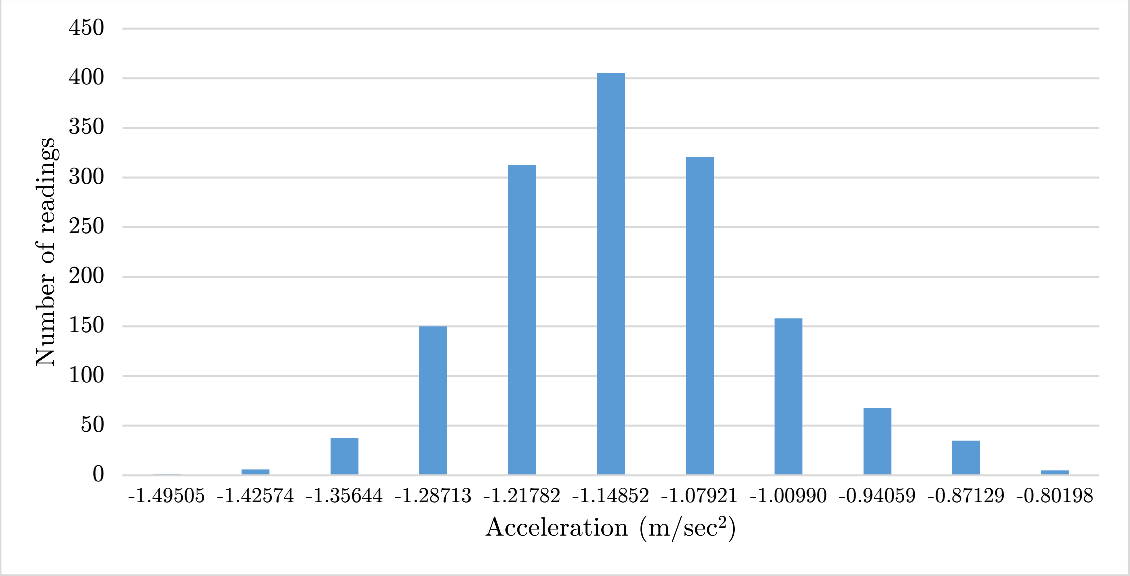

Figure 3.9. Calibrated x-axis acceleration histogram for sensor 3B79

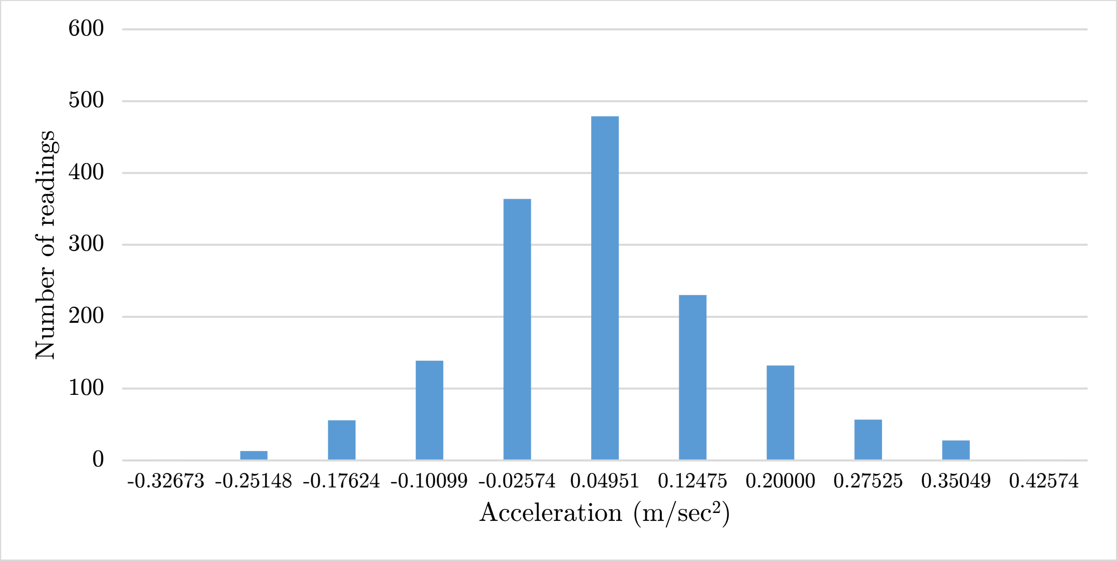

Figure 3.10. Calibrated y-axis acceleration histogram for sensor 3B79

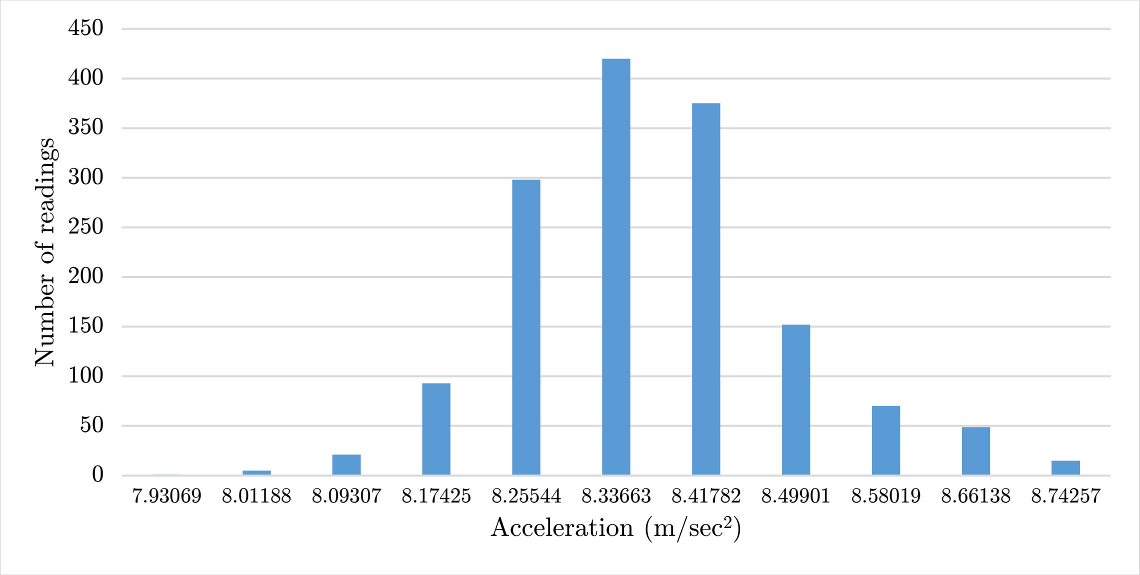

Figure 3.11. Calibrated z-axis acceleration histogram for sensor 3B79

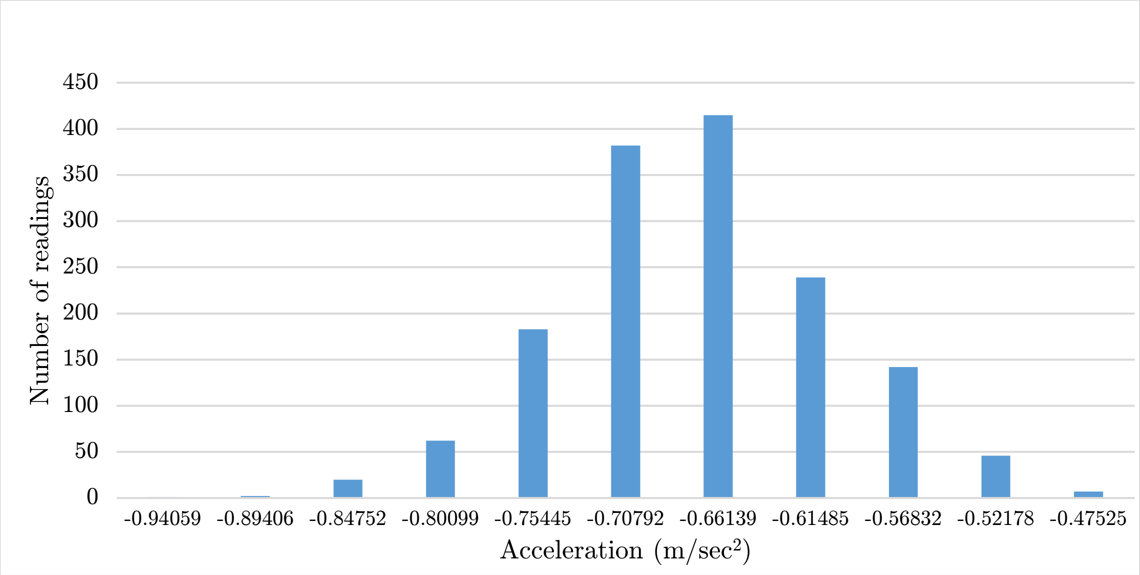

Figure 3.12. Calibrated x-axis acceleration histogram for sensor 3B81

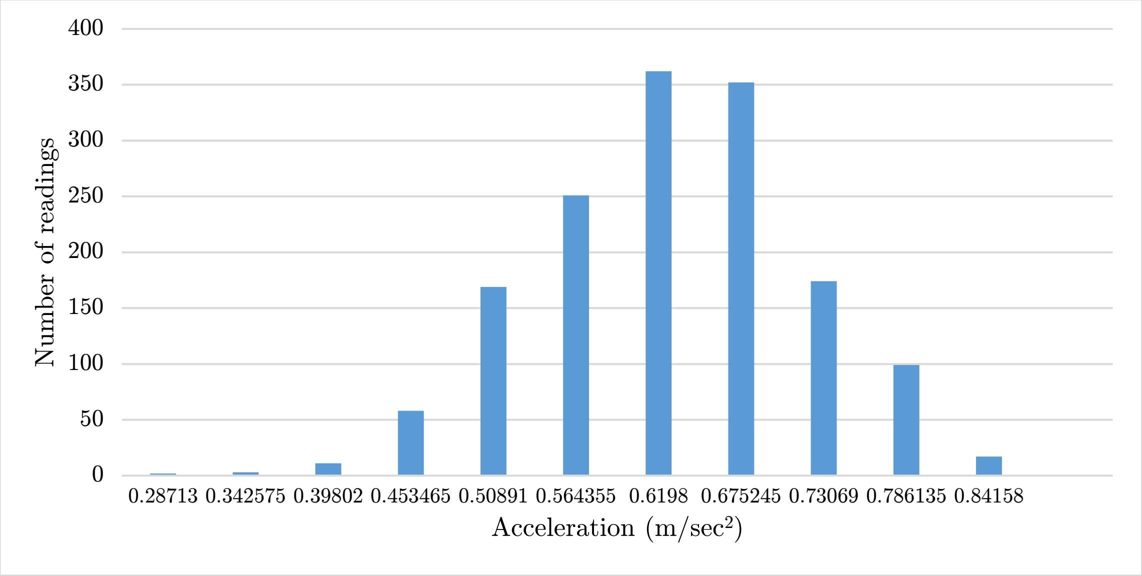

Figure 3.13. Calibrated y-axis acceleration histogram for sensor 3B81

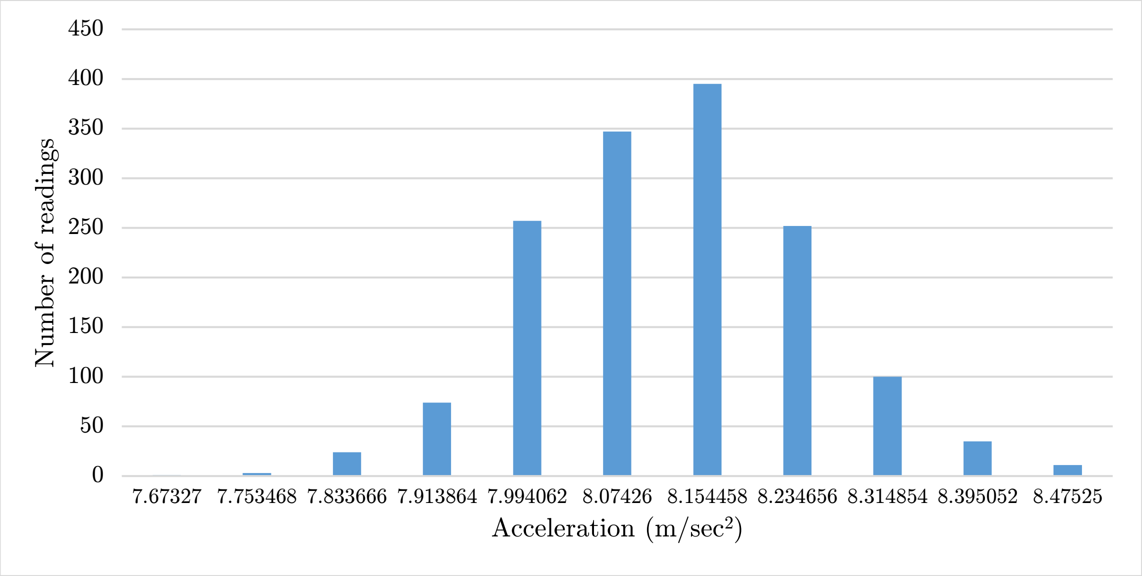

Figure 3.14. Calibrated z-axis acceleration histogram for sensor 3B81

The bias varies for each individual axis and no correlation has been seen between the individual axes or sensors. Figure 3.15 is a superimposed graph of individual accelerometer axis readings. The graph shows that each axis is unique and exhibits a unimodal distribution. Similar observations were obtained for the full set of Shimmer sensors used in the experiments.

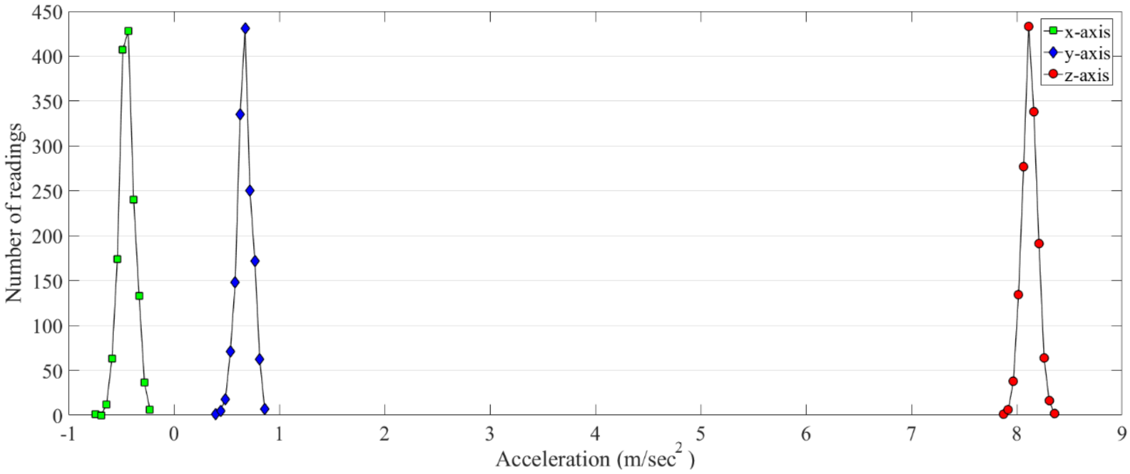

Figure 3.15. Superimposed accelerometer histogram for sensor 3B56

Statistical analysis of the accelerometer histograms proves that each sensor possesses a unique bias. The p-value also confirms that there is significant difference in readings obtained from the sensor. Given in table 3.2 – 3.4 is the statistical analysis of the readings obtained from three identical Shimmer accelerometers.

Table 3.2. Statistical analysis of 3B56 tri-axial accelerometer sensor

	Sensor 3B56		
	x-axis	y-axis	z-axis
Mean	-0.43432	0.64801	8.09927
Standard deviation	0.07083	0.07117	0.07093
Skewness	0.01077	-0.11039	0.04483
Confidence interval	-0.43791 to -0.43074	0.64442 to 0.65162	8.09568 to 8.10286
Kurtosis	0.01234	0.05141	-0.01994
p-value	0.00		

Table 3.3. Statistical analysis of 3B79 tri-axial accelerometer sensor

	Sensor 3B79		
	x-axis	y-axis	z-axis
Mean	-1.17590	0.00786	8.32627
Standard deviation	0.10649	0.11166	0.12268
Skewness	0.32939	0.30758	0.53298
Confidence interval	-1.18129 to -1.17051	0.00723 to 0.00849	8.32006 to 8.33248
Kurtosis	0.24464	0.42474	0.64909
p-value	0.00		

Table 3.4. Statistical analysis of 3B81 tri-axial accelerometer sensor

	Sensor 3B81		
	x-axis	y-axis	z-axis
Mean	-0.69565	0.59959	8.08433
Standard deviation	0.06769	0.08915	0.11850
Skewness	0.00793	-0.06997	0.11518
Confidence interval	-0.69908 to -0.69223	0.59508 to 0.60410	8.07833 to 8.09033
Kurtosis	0.08851	-0.13534	0.16224
p-value	0.00		

3.2.3 Gyroscope Bias Analysis

The Shimmer is embedded with a gyroscope sensor which measures rotations per second. To assess the bias in the gyroscope the sensor is subjected to a stimulus similar to that of the accelerometer. The stimulus for a gyroscope is applied by placing the gyroscope on a stable surface free from rotations and movements thus 0 deg/sec. This stimulus functions as a comparison point for all the readings obtained from the sensor. Under ideal conditions the readings from the gyroscope sensor should be equal to the stimulus. Repeated experiments show that this is not the case and that each axis of the gyroscope possesses a unique bias. The bias in the sensor is unique and reproducible provided the stimulus remains the same. The reading from the sensors confirm that there is sufficient statistical variances in the unimodal distributions obtained from the sensors. The statistical variances between different sensors makes the gyroscope bias an attractive implicit feature for ICMetric generation. Figures 3.15 - 3.23 show the calibrated gyroscope histograms obtained from three identical Shimmer sensors bearing identifications 3B51, 3B56, 3B79. From the graphs it is evident that each axis exhibits a different bias and that there is neither a similarity between the sensors nor a correlation between any axes.

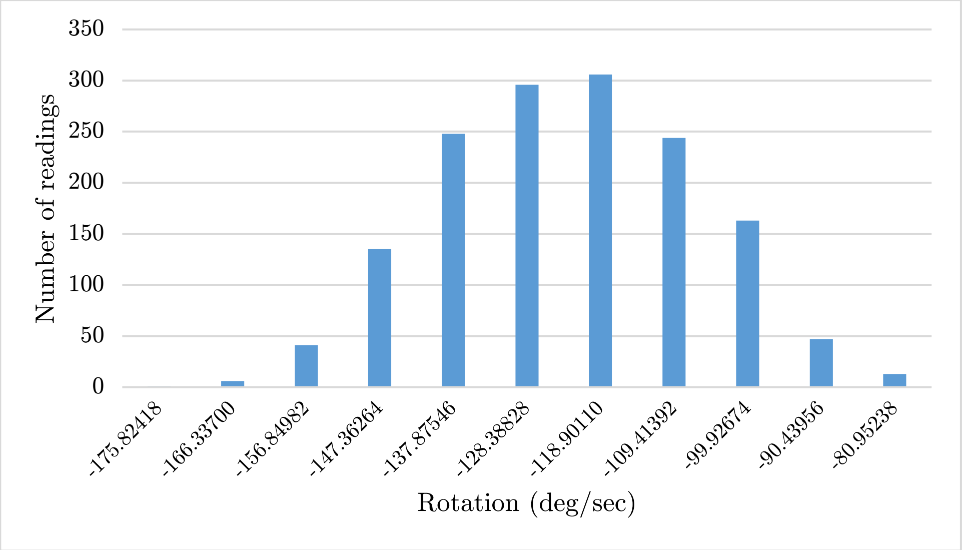

Figure 3.16. Calibrated x-axis gyroscope histogram for sensor 3B51

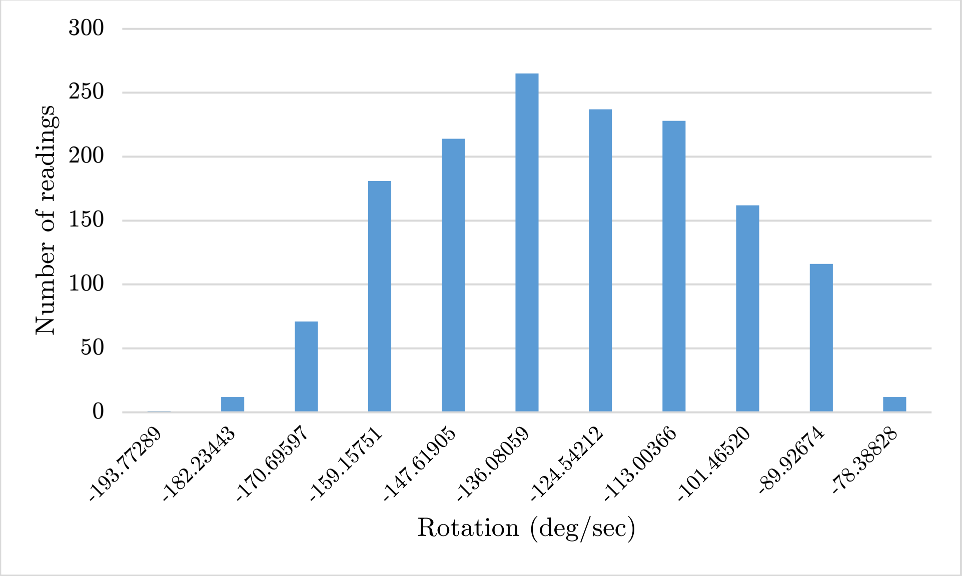

Figure 3.17. Calibrated y-axis gyroscope histogram for sensor 3B51

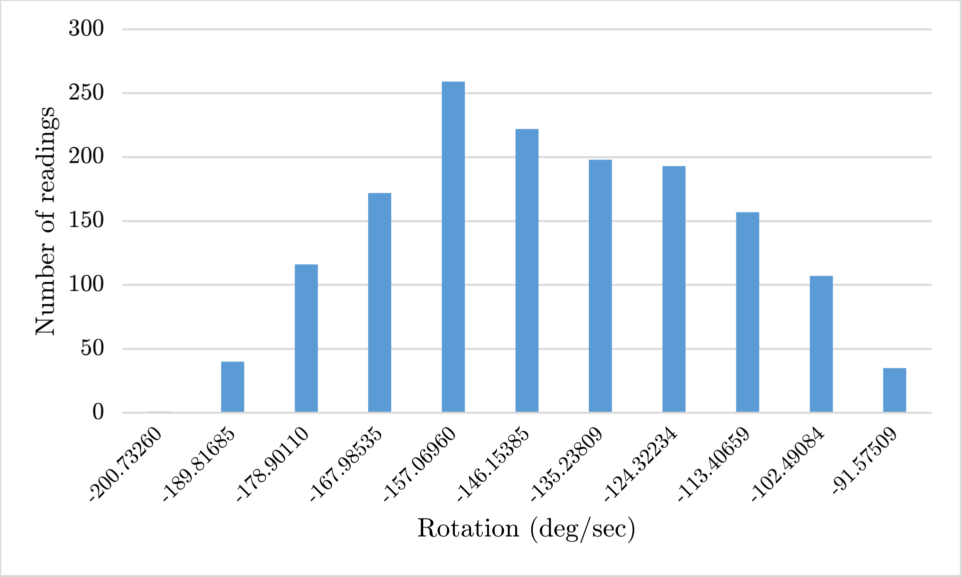

Figure 3.18. Calibrated z-axis gyroscope histogram for sensor 3B51

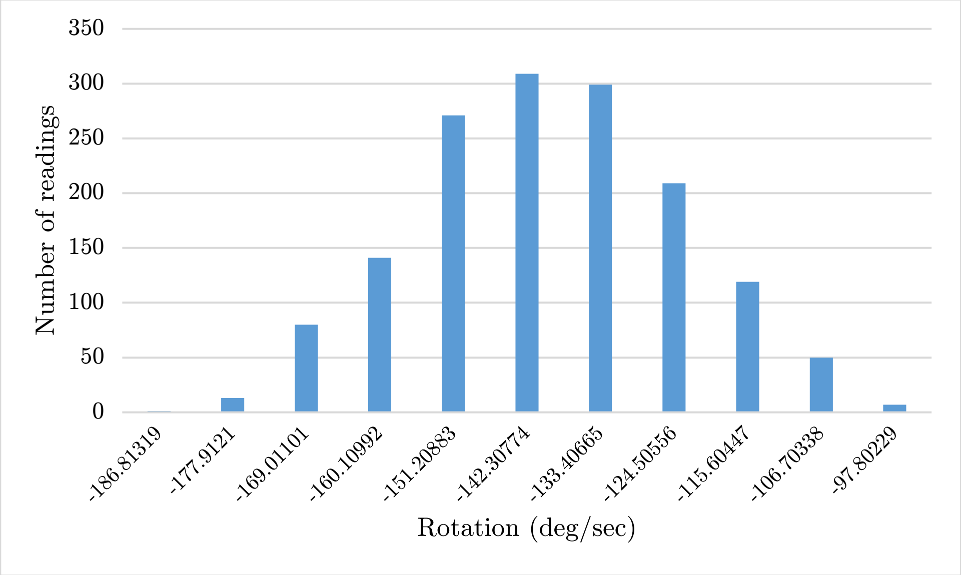

Figure 3.19. Calibrated x-axis gyroscope histogram for sensor 3B56

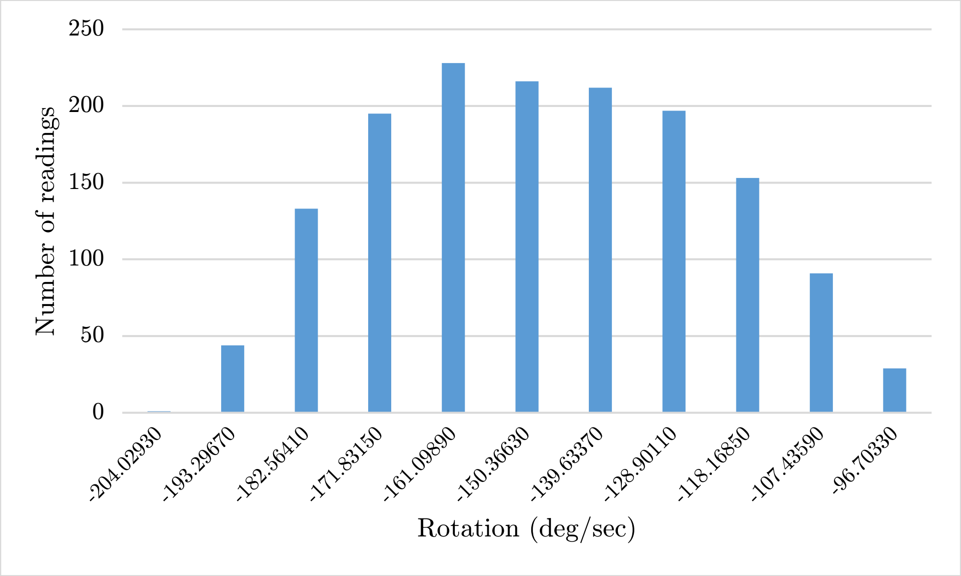

Figure 3.20. Calibrated y-axis gyroscope histogram for sensor 3B56

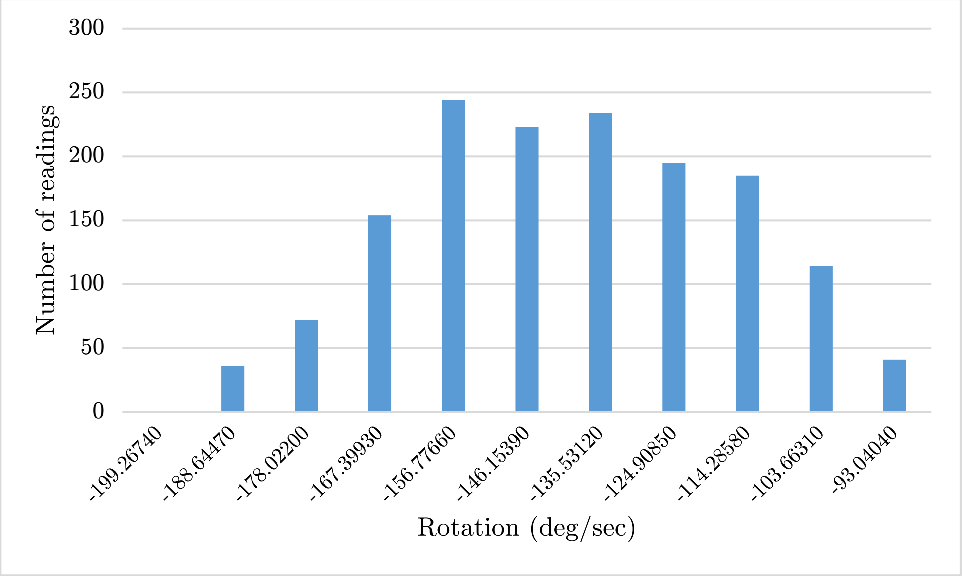

Figure 3.21. Calibrated z-axis gyroscope histogram for sensor 3B56

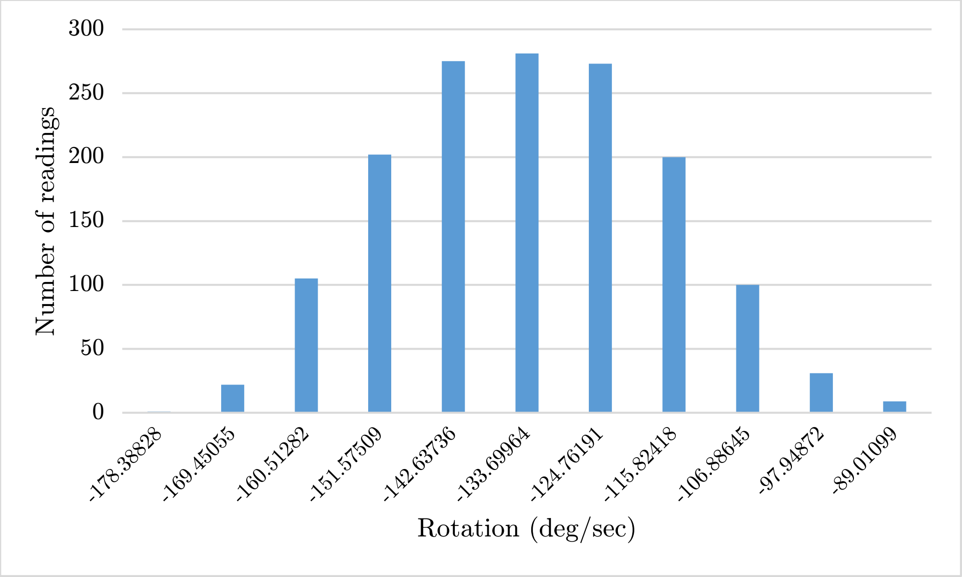

Figure 3.22. Calibrated x-axis gyroscope histogram for sensor 3B79

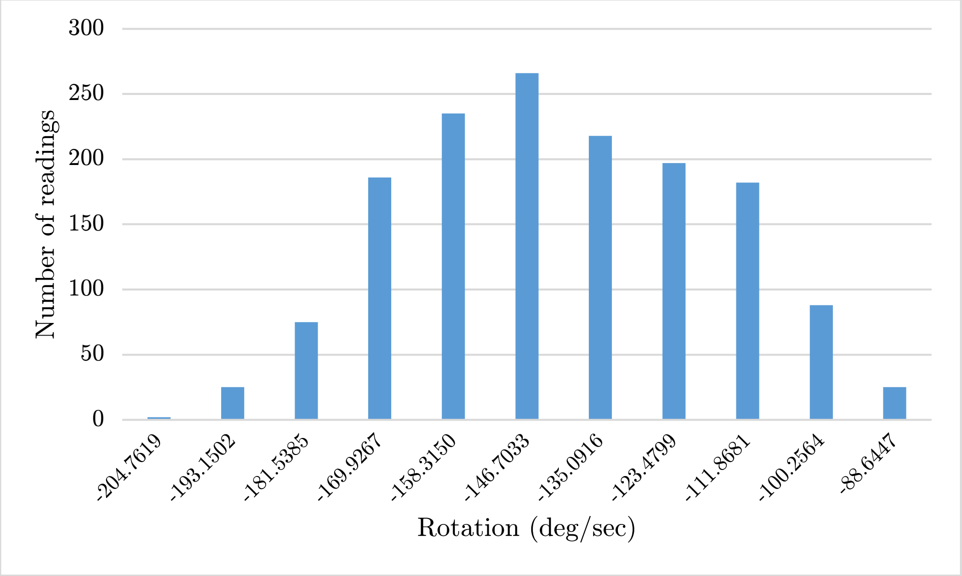

Figure 3.23. Calibrated y-axis gyroscope histogram for sensor 3B79

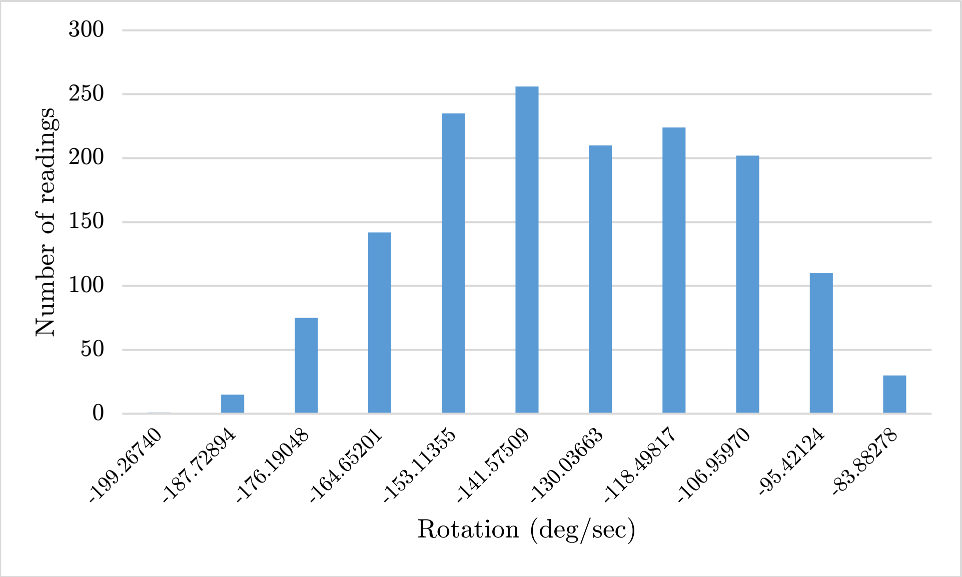

Figure 3.24. Calibrated z-axis gyroscope histogram for sensor 3B79

The bias varies for each individual axis and no correlation has been seen between the individual axes or sensors. Figure 3.25 is a superimposed graph of individual gyroscope axis readings. The graph shows that each axis is unique and exhibits a unimodal distribution. Similar observations were obtained for the full set of Shimmer sensors used in the experiments.

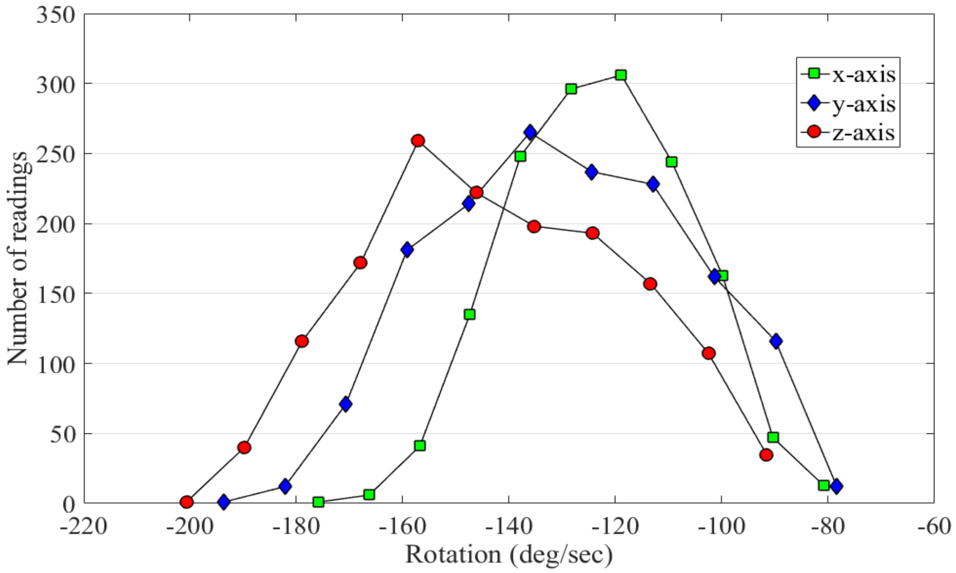

Figure 3.25. Superimposed gyroscope histogram for sensor 3B51

Statistical analysis of the gyroscope histograms proves that each sensor possesses a unique bias. The p-value also confirms that there is significant difference in readings obtained from the sensor. Given in table 3.5 – 3.7 is the statistical analysis of the readings from three identical Shimmer gyroscopes.

Table 3.5. Statistical analysis of 3B51 tri-axial gyroscope sensor

	Sensor 3B51		
	x-axis	y-axis	z-axis
Mean	-127.81196	-135.25128	-147.56654
Standard deviation	16.48308	23.20079	24.14309
Skewness	0.02725	0.02601	0.09415
Confidence interval	-128.64634 to -126.97759	-136.42570 to -134.07686	-148.78866 to -146.34442
Kurtosis	-0.44354	-0.83918	-0.79749
p-value	0.00		

Table 3.6. Statistical analysis of 3B56 tri-axial gyroscope sensor

	Sensor 3B56		
	x-axis	y-axis	z-axis
Mean	-143.95750	-152.98412	-145.04249
Standard deviation	15.84492	23.64065	22.81164
Skewness	0.03387	0.09183	0.00498
Confidence interval	-144.75958 to -143.15544	-154.18081 to -151.78744	-146.19721 to -143.88777
Kurtosis	-0.45522	-0.85886	-0.77233
p-value	0.00		

Table 3.7. Statistical analysis of 3B79 tri-axial gyroscope sensor

	Sensor 3B79		
	x-axis	y-axis	z-axis
Mean	-137.88913	-146.95286	-139.27032
Standard deviation	16.33432	23.80630	23.63067
Skewness	0.11677	0.07350	-0.00196
Confidence interval	-138.71597 to -137.06229	-148.15794 to -145.74780	-140.46651 to -138.07415
Kurtosis	-0.50523	-0.73155	-0.82448
p-value	0.00		

3.2.4 Strain Gauge Bias Analysis

The strain gauge sensor is by design a delicate sensor. The MEMS sensor is designed to detect strains that it experiences. When a strain gauge sensor is mounted or screwed into a plastic casing this could introduce a deformation which is exhibited in the readings. Under ideal conditions the strain gauge should accurately show the strain on the sensor. Thus if the sensor is left on a stable surface with no external influences then the sensor should show that no strains are being applied. Experiments on the strain gauge sensor show that the sensor possesses a bias which is reflected in the readings. The strain gauge in the Shimmer sensor is a dual polarity sensor. Hence the sensor provides as output two readings i.e. high and low polarity measured in mVolts. Figures 3.24 – 3.29 show the calibrated strain gauge histograms obtained from three identical Shimmer sensors bearing identifications 3B51, 3B81 and 3B4B. From the graphs it is evident that each axis exhibits a different bias and that there is neither a similarity between the sensors nor a correlation between any axes.

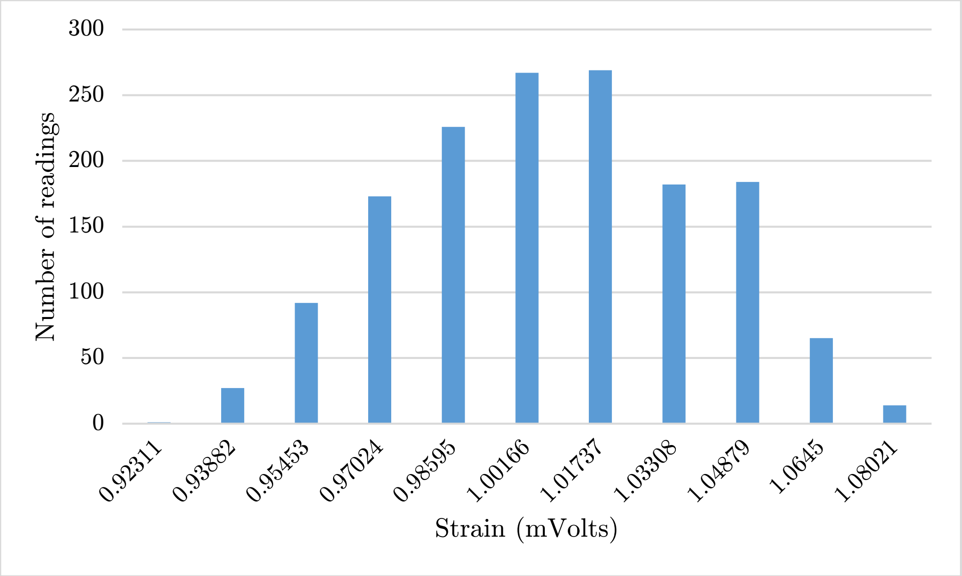

Figure 3.26. Calibrated strain gauge high polarity histogram for sensor 3B51

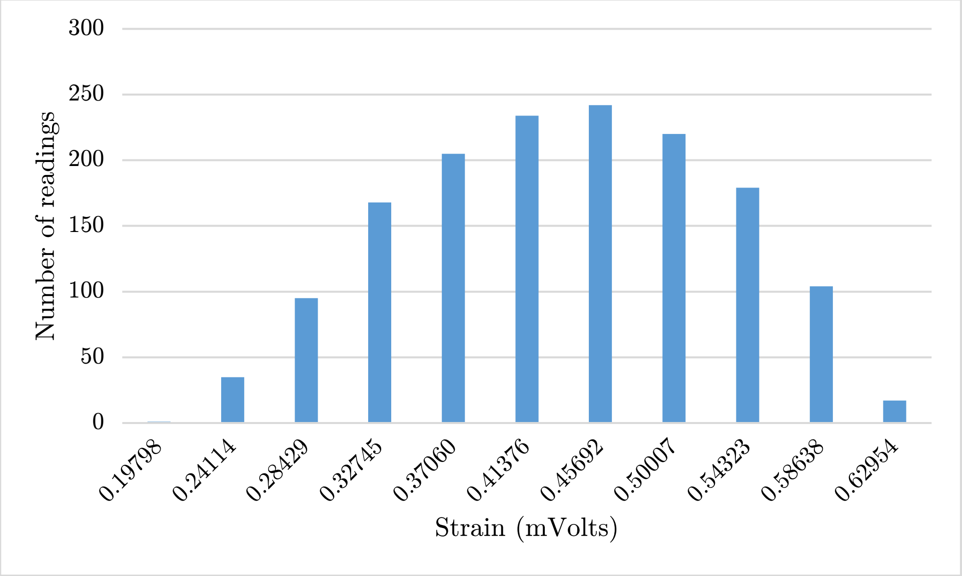

Figure 3.27. Calibrated strain gauge low polarity histogram for sensor 3B51

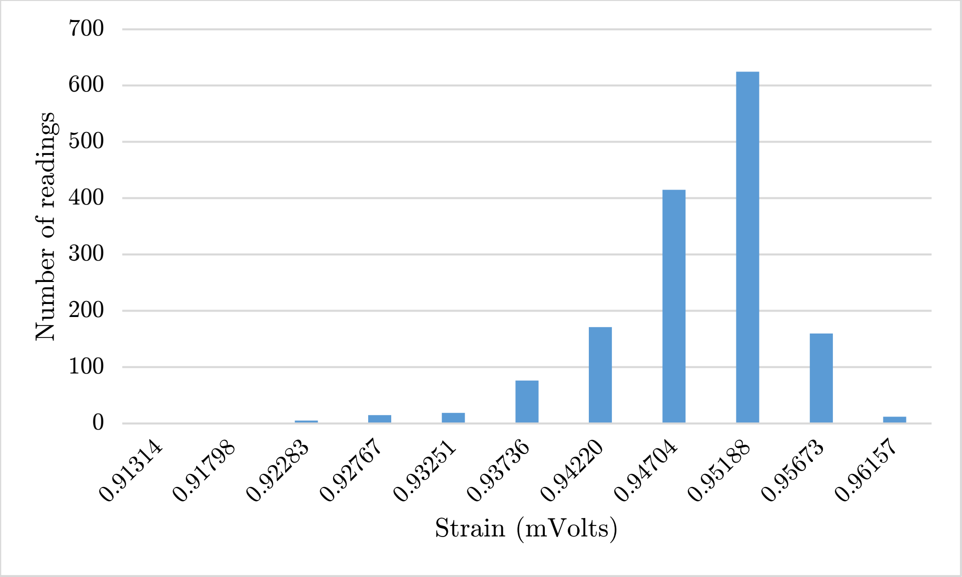

Figure 3.28. Calibrated strain gauge high polarity histogram for sensor 3B81

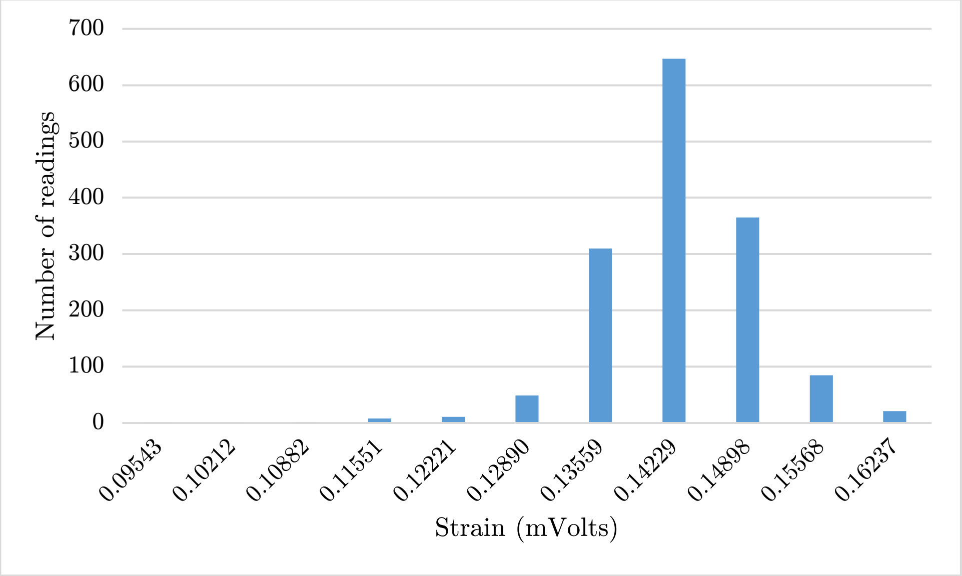

Figure 3.29. Calibrated strain gauge low polarity histogram for sensor 3B81

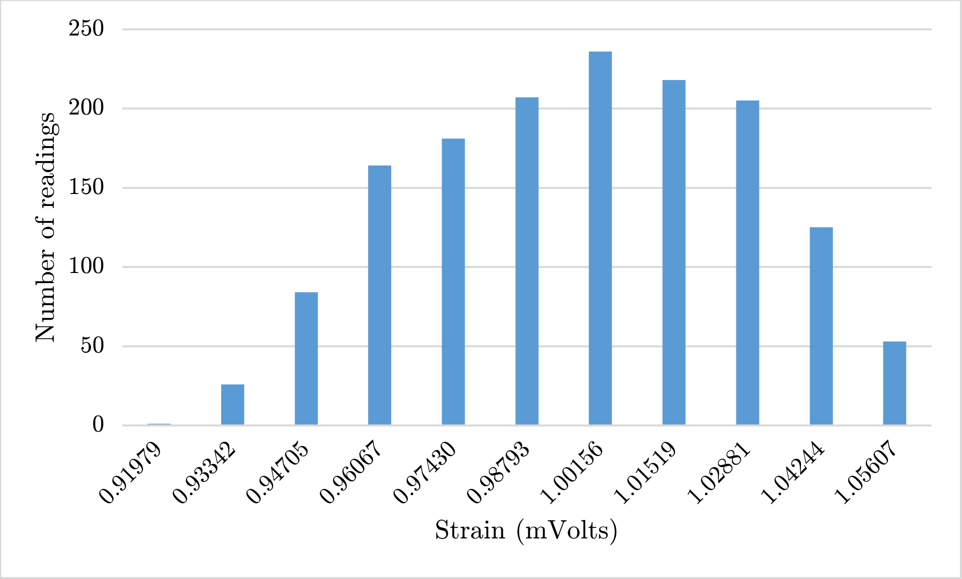

Figure 3.30. Calibrated strain gauge high polarity histogram for sensor 3B4B

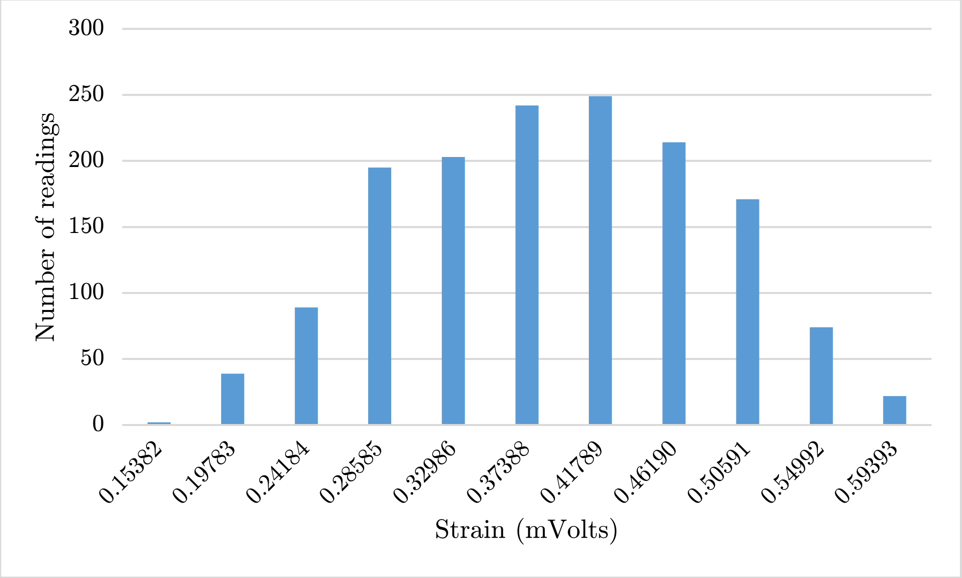

Figure 3.31. Calibrated strain gauge low histogram for sensor 3B4B

The bias varies for each individual axis and there is no correlation between two axes or sensors. Figure 3.31 is a superimposed graph of individual strain gauge sensor readings. The graph shows that each axis is unique and exhibits a unimodal distribution. Similar observations were obtained for the full set of Shimmer sensors used in the experiments.

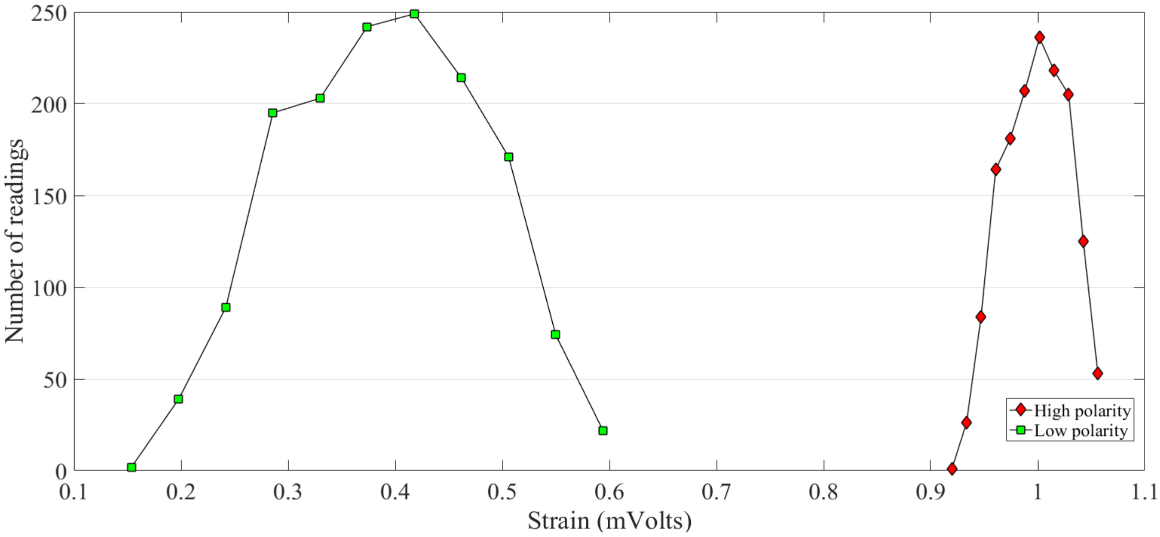

Figure 3.32. Superimposed strain gauge histogram for sensor 3B56

Statistical analysis of the strain gauge histograms proves that each sensor possesses a unique bias. Given in table 3.8 – 3.10 is a statistical analysis of readings from three Shimmer strain gauge sensors.

Table 3.8. Statistical analysis of 3B51 strain gauge sensor

	Sensor 3B51	
	High polarity	Low polarity
Mean	0.99965	0.41427
Standard deviation	0.03103	0.09056
Skewness	0.00846	-0.07965
Confidence interval	0.99809 to 1.00123	0.40969 to 0.41886
Kurtosis	-0.67545	-0.82690
p-value	0.00	

Table 3.9. Statistical analysis of 3B81 strain gauge sensor

	Sensor 3B81	
	High polarity	Low polarity
Mean	0.94635	0.13937
Standard deviation	0.00581	0.00679
Skewness	-1.28768	-0.48479
Confidence interval	0.94606 to 0.94665	0.13903 to 0.13972
Kurtosis	2.72895	4.19192
p-value	0.00	

Table 3.10. Statistical analysis of 3B4B strain gauge sensor

	Sensor 3B4B	
	High polarity	Low polarity
Mean	0.99176	0.36821
Standard deviation	0.03003	0.09164
Skewness	-0.09299	-0.00982
Confidence interval	0.99024 to 0.99328	0.36357 to 0.37285
Kurtosis	-0.83512	-0.75280
p-value	0.00	

3.3 SENSOR EXPLICIT FEATURES

A sensor like the Shimmer also possesses explicit features which can also be used for generating an ICMetric. Using only explicit features for the establishment of an ICMetric is a risk since these features may be easy to extract and spoof for an adversary. For instance the MAC address is often printed on the exterior of a device thus making spoofing an effortless task. This is precisely why the ICMetric should be based on a combination of explicit and implicit features. Using a combination of explicit and implicit features ensures that the resulting ICMetric is truly diverse. A comprehensive study on the explicit features possessed by the Shimmer sensor proves that the sensor is equipped with many unique features which are also possessed by other common devices and sensors. Given below are the features which can be used for generating an ICMetric:

Sensor MAC address - A unique 48bit MAC address associated with each sensor.

Bluetooth radio identification - A modifiable 16bit hexadecimal identity used to identify a device when connecting via Bluetooth.

Silicon serial identification - Each Shimmer sensor is embedded with a DS2411 chip [96] which is intended to provide a serial identification. The DS2411 is a single line factory lasered chip which is designed for equipment registration, peripheral identification, module identification and network node identification. Being a single line identification module ensures that the chip identification cannot be modified by an attacker. The DS2411 is composed of three unique code elements that can be incorporated into the device ICMetric. Figure 3.30 shows the DS2411 serialization structure:

CRC 8 bit	Serialization 48 bit	Family Code 8 bit

Figure 3.33. The DS-2411 serialization structure

Calibration matrices - The Shimmer sensor attempts to generate accurate readings using an offset, sensitivity and alignment matrix. These three matrices make corrections and adjustments to the readings obtained from a sensor. If an adversary attempts to inaccurately predict/modify the calibration matrices it will result in the wrong readings being obtained from the sensor. Hence even the slightest difference in the calibration matrices will result in a chain of events where both the sensor readings and the related calibration matrix are incorrect. Figure 3.31 shows the calibration matrices with sample calibration values:

-1
-1
-1

(a)

65535	0	0
0	65535	0
0	0	65535

(b)

-0.01	-0.01	-0.01
-0.01	-0.01	-0.01
-0.01	-0.01	-0.01

(c)

Figure 3.34. The calibration matrices with sample values (a) offset vector (b) sensitivity matrix (c) alignment matrix

3.4 KEYING ARCHITECTURES

Keying in groups is a delicate matter because no precise definition of a group architecture has been given in the literature. Often group architectures are compared with applications like social networking, chat applications (Skype, FaceTime) and group based services (google, yahoo groups). The comparison of security based group collaboration with a commercially available system is slightly flawed. Group secure communication architectures are recognized by how keying operations are performed in a group setting. Factors worth considering are whether keying is performed collaboratively or is dictated by the group controller. Below are the two keying architectures identified for secure group communication.

3.4.1 Dictative Keying

In the dictative approach, keying responsibilities are given to a group controller or to the Key Generation Centre (KGC). It is not necessary that the controller is a fixed entity. A client can be given privileges to carry out the role of a group controller. The dictative architecture is mostly based on the use of a dictated key that is communicated to the group members. The problem with this technique is that the controller needs to be protected from attacks because if the controller is compromised then the group communication is dismantled. Furthermore, a monolithic architecture can be more devastating if an attack is successful on the group. In its original form this architecture has limited coherence with the ICMetric technology because using such a setup would mean that a group key is generated without taking ICMetric inputs from the individual group members. In dictative keying the KGC can certify that the keys have been generated with a good random number generation source and that the keys possess the required properties. Figure 3.32 is a generic representation of the dictative keying architecture with the keys being communicated to the clients.

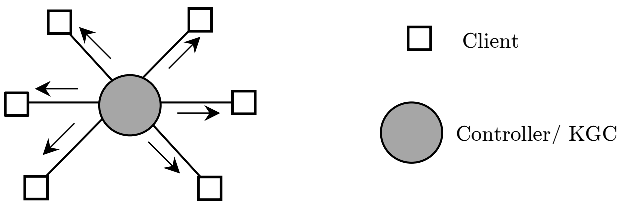

Figure 3.35. The dictative keying architecture with a KGC communicating with clients in the group

3.4.2 Contributive Keying

In contributive keying, clients are required to provide contributions upon which keys are generated. The group controller or KGC is responsible for performing a computation on the provided data so that a key can be generated. Once the key is generated it is then communicated to the individual group members which function as clients. A vast advantage of the contributive architecture is that it allows the generation of a contributive key which is generated by taking inputs from individual members in the group. Since this architecture requires key transportation therefore it is susceptible to the man-in-the-middle attack. Figure 3.33 is a generic representation of the contributive keying architecture with keys being generated collaboratively.

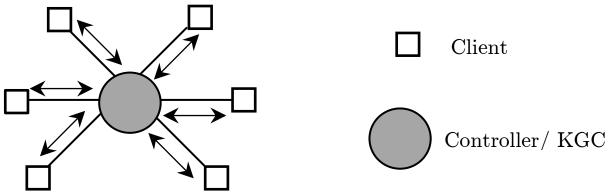

Figure 3.36. The contributive keying architecture with a KGC communicating with clients in the group

3.5 FRAMEWORK ASSUMPTIONS

A multiparty environment is composed of multiple devices communicating with each other. A challenge when providing security in a multiparty environment is the presence of dishonest participants. When using the ICMetric technology in the presence of dishonest participants, it is particularly important to ensure that the ICMetric of any device or the group is not exposed.

A multiparty communication system is composed of multiple computation devices connected remotely. To administer the service of the group the proposed schemes require a KGC. The KGC plays an important role in a group setting because it is responsible for providing access when a device wishes to join the group, issue a new key to the group and perform rekeying whenever a device leaves the group. It is assumed that the devices in the group and the KGC are ICMetric capable. To protect from eavesdropping it is assumed that all communications are performed via secure channels. Figure 3.34 is a pictorial representation of the secure group communication. The various form of wearable and ubiquitous devices connects to the KGC to establish secure group communications.

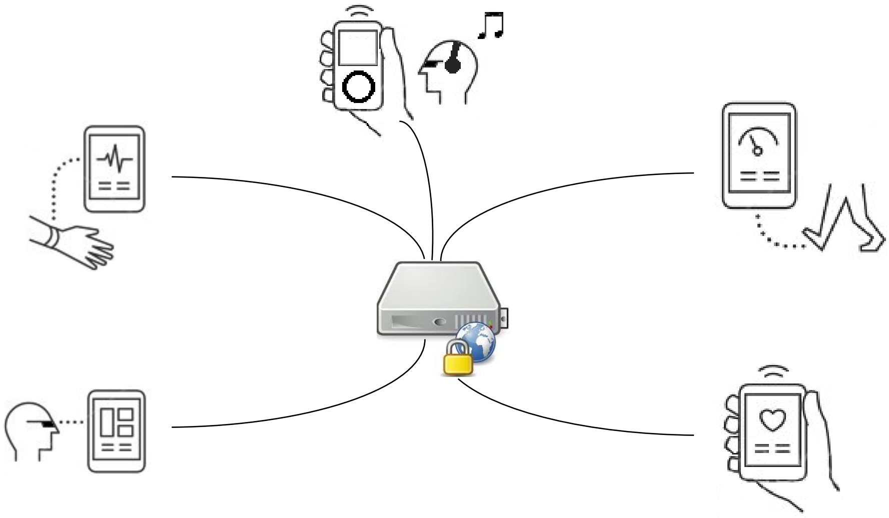

Figure 3.37. Basic system model for secure group communications

3.6 CREATING A GROUP ICMETRIC

An ICMetric identifies a single device based on its internal environment. A similar identification is required to identify a group of devices which are communicating in a secure group environment. This identification, called the group ICMetric uniquely identifies the group and must be generated using the ICMetric of the individual devices that form the group. Besides providing an identification to the multiparty environment, the group ICMetric can also be used to administer cryptographic services. Generating the group ICMetric is a challenging task because it has to be generated using the individual device ICMetric. Besides this the following requirements make computing the group ICMetric an even complex task:

- The ICMetric of any device must not be communicated to even trusted entities in the group.
- The group ICMetric must not be communicated to any entity outside the group.
- An adversary should not be able to recover the individual ICMetric of any device that forms the group.

To generate the group ICMetric without exposing the individual ICMetric Shamir's Secret Sharing scheme is used. This scheme allows a number of individual members/ devices to construct the secret group ICMetric.

To generate the group ICMetric the devices in the group will be sent a temporary salt s_{temp}. This will be used by the device to generate a hash by adding its own ICMetric icm_d and the s_{temp} as follows:

$$ich = hash(icm_d + s_{temp}) \tag{7}$$

Each device in the group will send its ID and ich to the KGC. Thus the KGC will maintain ID and ich pairs i.e. $(ID_x \, , ich_x)$. The responses obtained from the individual devices will be used to form the secret share points required for constructing the group ICMetric.

$$\{(ID_1, ich_1), (ID_2, ich_2), (ID_3, ich_3), \cdots, (ID_t, ich_t)\} \tag{8}$$

The secret group ICMetric is constructed by using Lagrange interpolation. Lagrange polynomial is used with the previously assembled share points. The group ICMetric is assembled using the following polynomial:

$$icm_g = \prod_{j=1}^{t}(ID_j) \sum_{i=1}^{t} \frac{ich_i}{(ich - ich_i)\prod_{j=1,j\neq i}^{t}(ich_i - ich_j)} \qquad (9)$$

Where t is the number of individual pairs used for establishing the group ICMetric.

3.7 IMPLEMENTATION AND OUTCOMES

The accelerometer, gyroscope and strain gauge sensors in the Shimmer sensor have a sampling rate of 51.2Hz. Once the readings are obtained from the Shimmer sensor in a CSV file the readings are used to perform a statistical analysis of the readings of the sensor. The statistical analysis of the generated CSV file is performed in MATLAB. MATLAB takes 0.412 seconds to statistically analyse the CSV file using the accelerometer, gyroscope and strain gauge MEMS sensors. Experiments show that the sensors possess a unique repeatable bias. Statistically each sensor possesses unique characteristics which proves that the features can be used for ICMetric generation.

3.8 FEATURE STABILITY

The ICMetric technology and PUF rely heavily on the accurate extraction of device features. If the extracted feature readings are contaminated then the resulting device ICMetric and PUF methods will fail. Owing to this PUF research has tried to identify features that exhibit stability along with a reduced bit error rate [125]. There are limited experiments on the ICMetric technology owing to which research needs to be conducted on the effect of feature instability on ICMetric generation. Previous studies [90][91][92] have identified hardware features which can be used for ICMetric generation. Experiments show that the

Program Counter (PC) and Cycles Per Instructions (CPI) can used to generate an ICMetric.

MEMS sensors are mechanical components owing to which they are susceptible to external factors which can be difficult to simulate in a laboratory. MEMS sensors can be affected by the humidity [126], operational temperature and physical shock [127][128]. Being mechanical in nature means that the moving components will wear over time and their resulting behaviour will change. MEMS sensors can be based on many individual mechanical components like springs, combs and shuttles. Each MEMS component will damage differently for instance springs will buckle or misalign while the conductor foil in a strain gauge can buckle permanently [129]. Mechanical and structural properties change when a MEMS sensor is mishandled. Hence dropping a sensor or subjecting it to abnormal conditions (fatigue, stress and strain) can damage a sensor permanently.

If a sensor is studied minutely under a microscope numerous imperfections can be noticed as the device is subjected to everyday conditions. For instance particle contamination, debris, human hair, broken pin joints and missing linkages have been documented in literature [128]. Similarly a factor influencing the performance of all MEMS sensors is the presence of stiction. Any object that is in contact with another surface will require a minimum threshold to overcome static cohesion. If a sensor is unable to overcome stiction, it will not be able to respond to a supplied stimulus correctly [130].

Stability of features is a necessary requirement for generating the device ICMetric. If a sensor does not function as intended then it will result in the wrong ICMetric being generated. This will eventually lead to authentication failure and wrong cryptographic keys being generated. As this is beyond the scope of this research therefore it is worth exploring how the aging of a MEMS device influences the ICMetric of a device. Perhaps owing to their small size the effect of temperature was not prevalent in the Shimmer sensor. The same cannot be said for powerful and computationally complex devices.

3.9 SUMMARY

The ICMetric technology has been designed as a method of deterring cryptographic key theft. By design the ICMetric is an identification which can be used to identify a device based on its internal environment. Key generation, authentication and many other cryptographic services will be based on the ICMetric of a device therefore it is important to use features that are unique, repeatable and reproducible. This chapter presents a study on both explicit and implicit features that can be used for the creation of a device ICMetric. Modern embedded systems are often equipped with MEMS sensors like the accelerometer, gyroscope and strain gauge. Although these are precision sensors they possess a bias which can be seen in the readings obtained from the sensors. Experiments on the body wearable Shimmer sensor show that each sensor possesses a unique bias which can be analysed statistically. This chapter statistically studies the readings from embedded MEMS sensors to prove that there is a unique bias in every sensor. Experiments show that the sensor bias is an implicit feature which can be used for ICMetric generation. To strengthen the ICMetric, explicit features can also be used with implicit features. This chapter explores a range of explicit features which can be used like MAC address, identifications, calibration matrices etc.

The chapter proposes a system architecture where multiple devices connect to a single group controller. This group controller is an intermediary which is responsible for establishing the secure group communications. A goal of this research is to establish secure group communications using the ICMetric technology. This chapter proposes a scheme which uses the individual device ICMetric to create a group ICMetric. The group ICMetric is assembled using Shamir Secret Sharing scheme and holds the same properties held by the individual device ICMetric. The group ICMetric forms a secret identification for the group which can be used for the creation of symmetric and asymmetric keys for the group.

CHAPTER 4

SYMMETRIC KEY BASED GROUP COMMUNICATION

Secure group communication requires the establishment of a cryptographic key which can be used for a range of cryptographic services like encryption/ decryption. In the lifecycle of a cryptographic key the key generation is the most important phase. A cryptosystem can be considered weak if the key generation, key exchange or the key storage is flawed. Creating a cryptographic key is a challenging task especially in the multiparty environment because keys need to be generated securely and efficiently. Two types of keying exist in cryptography i.e. symmetric key and asymmetric key. This chapter explores the creation of a symmetric key for a multiparty environment using the ICMetric technology. The group ICMetric cannot be used as a cryptographic key, therefore schemes are required which can produce a cryptographic key from the group ICMetric. A challenge while generating the symmetric key is that the cryptographic key should be generated using the group ICMetric but this should not compromise the security of group or its members. Hence the proposed algorithm builds on sound cryptographic principles and ensures that the properties of the ICMetric technology are not violated in any way. The proposed algorithm is based on Password Based Key Derivation Function. The function uses a number of atomic primitives like cryptographic salts, hashing, and a large iteration count to create an ICMetric based symmetric key. The greatest advantage of the symmetric key generation algorithm is that it is adaptable to varying key sizes. The symmetric key generation scheme has been simulated and tested for varying key sizes and iteration count. The chapter first introduces the primitives of the scheme after which the symmetric key generation scheme is presented. The chapter concludes with the simulation details and results.

4.1 SYMMETRIC KEY CRYPTOGRAPHY

Symmetric key algorithms use a paradigm where a single key is used for the provision of cryptographic services. Hence the symmetric key forms a shared secret between two or more parties. To successfully execute encryption and decryption all parties involved must use the same symmetric key. If Alice and Bob wish to share a secret they agree on a symmetric key. When Alice wishes to send a message (plaintext) she encrypts the message using the symmetric key. The resulting ciphertext is transmitted to Bob, who will use the symmetric key to extract the plaintext from the ciphertext. Clearly, any person who has access to the symmetric key can also perform the decryption. Figure 4.1 shows the working of a symmetric key for the provision of encryption and decryption.

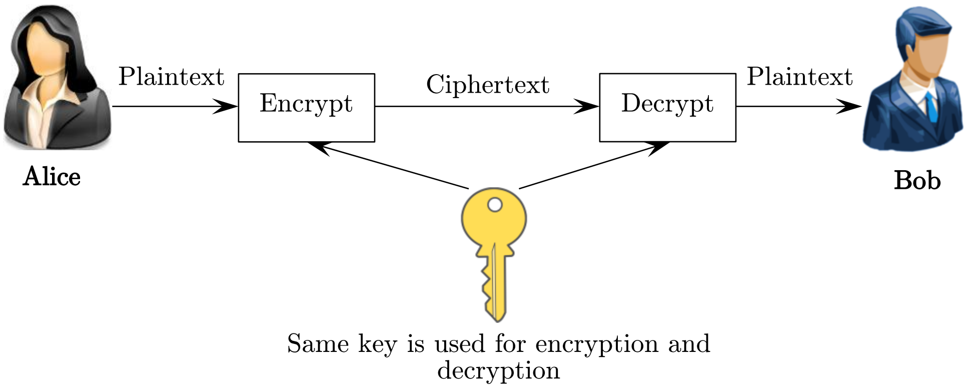

Figure 4.1. Encryption-decryption process using a symmetric key

Definition 4.1. For encryption and decryption transformations $\{E_e: e \in \mathcal{K}\}$ and $\{D_d: d \in \mathcal{K}\}$ where $\mathcal{K}$ is the key space, then under symmetric key cipher $e = d$ [22].

The symmetric key has an advantage that it is fast and a single key can be shared between a large number of individuals in a group setting. A concern while using symmetric keys is that if the key generation or key exchange process is compromised then the entire system is also compromised. Therefore a challenge with symmetric keys is communicating the key to the individual parties and then ensuring that the key remains secret.

4.2 SCHEME PRIMITIVES

The symmetric key generation scheme is based on a number of cryptographic primitives. Before providing a detailed description of how the symmetric key is generated it is important to throw light on the scheme primitives.

4.2.1 Password Based Key Derivation Function

A password is a string of characters chosen by a user to prove authentication so that access to a resource can be provided. A password on its own does not possess sufficient entropy owing to which it cannot be used as a cryptographic key. System designers attempt to increase the entropy of the password by suggesting an increase in the length of the password and also by suggesting incorporation of special characters and symbols. In many applications such as protecting data in a storage device the password is the only secret information upon which cryptographic services can be based. In such applications a method is required that takes as input a password and provides as output keying material which can be used for the provision of cryptographic services like authentication, access control etc.

Key Derivation Function (KDF) address the need for cryptographic key generation based on secret input. A KDF is a deterministic algorithm which is used to derive a cryptographic key based on secret information. A KDF can also be used to stretch or reduce the length of a key so that it conforms to the requirements of a cryptosystem. The derived keys from a KDF possess qualities like sufficient entropy, length and irreversibility owing to which KDF are an attractive tool in cryptography.

A Password Based Key Derivation Function (PBKDF) is a KDF that takes as input a password and produces as output a cryptographic key. The PBKDF uses salt based hashing and a large iteration count to produce a symmetric key which can resist rainbow table attacks. Hence the input of a PBKDF is the secret input, salt and key length that produces a symmetric key of the desired key length. Figure 4.2 shows the generic diagram of a PBKDF.

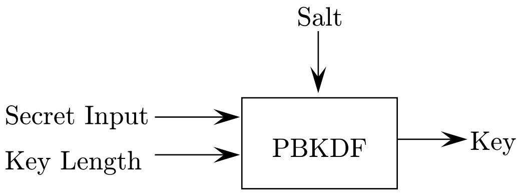

Figure 4.2. Generic flow diagram of PBKDF

The secret input of PBKDF bears close resemblance to the properties of the ICMetric of a device. For instance the password is a secret phrase which cannot be transmitted, similarly the ICMetric is secret and cannot be transmitted. A quality of the PBKDF is that the large sized iteration count prevents an attacker from extracting the password from a key, thus preventing reversal.

4.2.2 Cryptographic Salt

Many security primitives are based on a source of randomness. Often this source of randomness is obtained by incorporating a salt into the scheme. In cryptography a salt is used as an additional input to a one way function. It is mostly added as input to a hash function to make it difficult for an attacker to crack the output of the function [131]. This is achieved by incorporating random data (salt) so that the output hash can also be randomized. This property is particularly important in situations when a single input needs to be hashed in different instances to produce different hashes every time.

The PBKDF algorithm requires a salt to operate. The purpose of a salt here is to defend against dictionary attacks and computed rainbow table attacks. The salt used in PBKDF must be at least 128 bits. By using PBKDF, a new key can be generated for every salt value even if the iteration count and secret input remain the same. This property makes it difficult for an attacker to generate a table of possible keys. For a given input password the number of possible keys is 2^{sLen}, where $sLen$ is the salt length. The proposed ICMetric based symmetric key scheme uses a 128 bit salt.

4.2.3 Hashed Message Authentication Code

When transmitting or storing data in an insecure environment, the parties involved would want guarantees about the authenticity of the source and integrity of the data. To achieve this, Message Authentication Codes (MAC) use a shared secret key to generate a small data block that can be appended to the original message and sent. The data block is obtained by running the message through a MAC generation algorithm. When the receiver receives the message he computes the same MAC as a function of the original message. If the received and the computed MAC differ then it can be concluded that the message has been altered in some way. Since the secret key is only known to trusted parties therefore an attacker cannot alter a message and its associated MAC. This scheme ensures both authenticity of the sender and the integrity of the message.

A MAC is based on a symmetric block cipher that lacks efficiency. Therefore a method is required that provides improved performance and portability. An HMAC [132][133] is a keyed hash MAC which uses hashing instead of block ciphers to achieve improved performance and portability. Hash functions are typically faster than block ciphers and an HMAC is designed to work with all variants of SHA and MD. The HMAC is a lightweight algorithm which follows the same principle as that followed by a MAC. In an HMAC the key and message are both hashed to create a data block which can be appended to the original message and transmitted to the other party. Upon receipt the receiver will compute a hash on the message and the secret key. Equivalence of the computed HMAC and the received HMAC ensures that the message has not been altered and that the party sending the message is authentic. An HMAC is also preferred over a simple MAC as the appendable code is not intended to be encrypted. This provides justification for using the hash as an appropriate building block in a scheme.

4.2.4 Iteration Count

The iteration count is a numeric value which defines how many iterations are performed to generate the key. The iteration count is intended to make it difficult for an adversary to capture the keys of a system. Doing so also has an influence on the amount of computation required to generate the key for a legitimate user. While choosing the number of iterations it is important to establish a quantity which has low impact on user perceived performance but makes it difficult for an attacker to break the system. The NIST standard [134] targeting PBKDF suggests a minimum of 1000 iterations with an increased count where possible. The standard also recommends an iteration count of 10,000,000 may be appropriate in situations where the user perceived performance is not critical. There are many applications where a higher iteration count has been recommended [135] and experimented with. For instance, Apple iOS 9.0 uses PBKDF with 10,000 iterations [136] for the iTunes application. As CPU power becomes increasingly inexpensive it has been recommended that the iteration count should increase yearly at perhaps 40%-60% [135]. Choosing the number of iterations is dependent on the amount of resources available and the capability of the target computation device.

4.3 ICMETRIC BASED SYMMETRIC KEY GENERATION

As the ICMetric lacks size, entropy and necessary properties therefore it cannot be used as a cryptographic key [93]. Hence a scheme is required which allows ICMetric based secure symmetric key generation in a multiparty environment. The proposed scheme is intended to secure devices communicating in a multiparty environment. The scheme is founded on the ICMetric technology and composed of five different phases i.e. device imprinting, device authentication, group ICMetric generation, symmetric key generation and stream confidentiality. Table 4.1 provides a description of the symbols and variables used in the scheme.

Table 4.1. Symbols and variables used in the scheme

Symbol	Meaning
$\oplus$	Bitwise exclusive OR
$\|$	Concatenation
$\lceil\ \rceil$	Ceiling function
ich	Hashed ICMetric
s	128 bit random salt
icm_d, icm_{KGC}	ICMetric of device and KGC respectively
icm_g	ICMetric of the group
ID_x	Identity associated with device x
C	Iteration count; minimum 1000
$kLen$	Length of master key in bits
$hlen$	Digest size of hash function
$be(x)$	32-bit encoding of integer x. Significant bit appears on left
mk	Master key

4.3.1 Device Imprinting

The first step for establishing the group is imprinting [137]. The step is aimed at resolving the issue of establishing trust between two distrusting devices. The process mimics the duckling imprinting phase where a newborn duckling establishes a pattern with its parents.

When a device wishes to join a group it will first need to register with the KGC. The KGC is responsible for coordinating and supporting the presence of the group and its individual members. When a device wishes to join the group, the device will compute a hash of its ICMetric icm_d and send it to the KGC as follows:

$$h = hash\ (icm_d) \tag{10}$$

This value of h is discarded by the device but is stored by the KGC for future authentication.

4.3.2 Device Authentication

To prove authenticity, the device will compute a hash of h and a temporary salt s_{temp} issued by the KGC. The device will respond by computing:

$$h_1 = hash(h + s_{temp}) \tag{11}$$

The KGC will compute the same and compare the resulting value with that provided by the device. If both values are identical then the device will be authenticated. Upon successful authentication each device will be allocated a unique identity ID, which will help in establishing the group ICMetric as outlined in chapter 3.

4.3.3 Password Based Symmetric Key Derivation

The symmetric key generation scheme uses PBKDF with the group ICMetric icm_g to generate a master key mk of length $kLen$ which is used for confidentiality services. The key length $kLen$ can be modified to conform with the needs of the encryption algorithm utilized for confidentiality. The proposed algorithm also takes as input the digest length $hLen$. A modifiable key length and an adaptable HMAC scheme allows the algorithm to conform to changing cryptographic requirements. The flow diagram for symmetric key generation is given in figure 4.3.

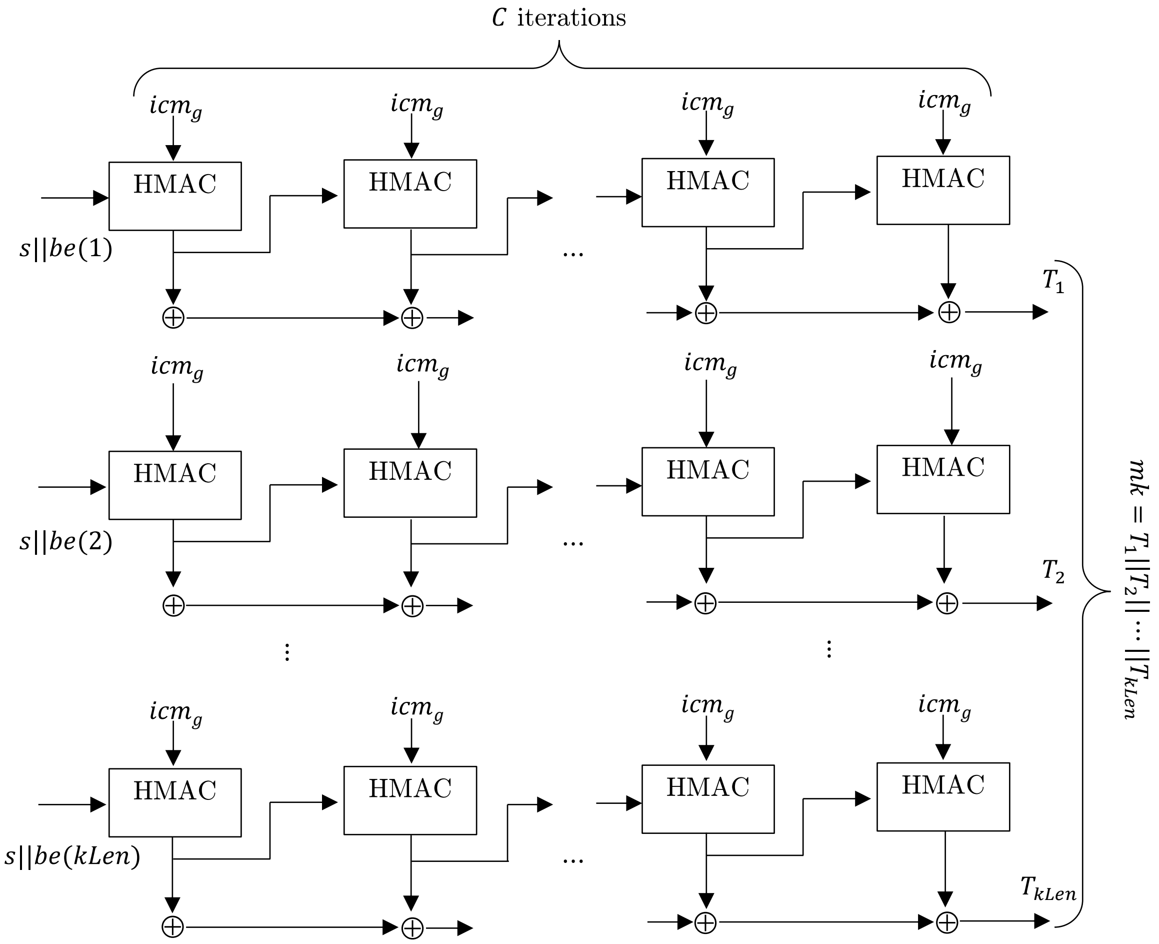

Figure 4.3. The PBKDF schematic showing the generation of a symmetric key using the group ICMetric icm_g

Since all involved parties and the KGC possess the same group ICMetric therefore it can be concluded that using the group ICMetric will result in a single symmetric key for all parties in the group. The PBKDF algorithm takes as input the password (group ICMetric), salt, iteration count and required key length. The symmetric key generation algorithm has a constraint that the length of the final key should be at most $(2^{32} - 1) \times hLen$.

```
If (kLen>$(2^{32}-1)\times$hLen)
    Return with error
len=$\lceil$kLen/hLen$\rceil$
r=kLen-(len-1)$\times$hLen
for (i=1 to kLen)
{
    $T_i$=0
    $U_0$=s $\|$ be(i)
    for(j=1 to C)
    {
    $U_j$=HMAC($icm_g$,$U_{j-1}$)
    $T_i$=$t_i \oplus U_j$
    }
}
Return mk=($T_1\|T_2\| \cdots \|T_{kLen}$)
```

This ICMetric based symmetric key algorithm creates a symmetric key *mk* of length *kLen* for the group. The key can be used for encrypting and decrypting messages that are being communicated in the group setting.

4.4 IMPLEMENTATION AND OUTCOMES

The proposed symmetric key generation scheme has been implemented and tested on Intel Core i5 3.4GHz processor computer with 6GB RAM. The MEMS readings are obtained from the Shimmer sensor while authentication, group ICMetric generation, symmetric key generation and confidentiality schemes have been implemented in Bloodshed Dev-C [138] and MATLAB. Cryptographic functionalities are provided by the OpenSSL cryptographic library [139].

The proposed system is composed of four subcategories each targeting a different components. The system is composed of the following modules:

- ICMetric generation – A module dedicated to creating a group ICMetric from ICMetric of the group devices.
- Authentication – A module designed to authenticate the individual devices in the group environment.
- Key generation – A PBKDF scheme that generates keys of varying sizes i.e. 128, 256, 512, 1024 bits.
- Confidentiality – Two stream cipher modules i.e. Rabbit stream cipher and AES (128 and 256 bit).

4.4.1 Outcomes

Once the ICMetric has been generated, the device will use authentication services to get itself authenticated using its ICMetric. The authentication scheme contains two occurrences of the SHA256 function and takes 5×10^{-3} seconds to run. The group ICMetric generation is based on Shamir Secret Sharing Scheme. This module requires 1.5×10^{-3} seconds to run top-down. Figure 4.4 shows the time required for statistical analysis in ICMetric generation, authentication and the group ICMetric generation.

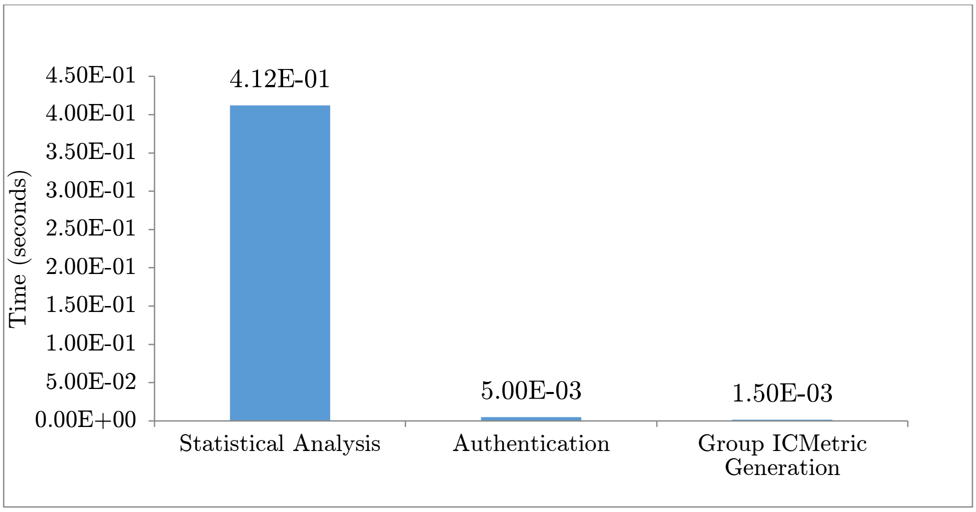

Figure 4.4. Graph showing time (seconds) taken by the modules of the scheme

The PBKDF is influenced by two parameters i.e. the key size and the number of iterations. To study the performance of this algorithm four common keys were generated using varying number of iterations and key sizes. The generated key sizes were 128, 256, 512, 1024 bits while the tested iteration count is 1000, 2000 and 4000. Table 4.2 shows the time taken by the PBKDF when subjected to varying key sizes and iteration count.

Table 4.2. Time taken by PBKDF with varying key size and iteration count

		Key Size (bits)	Key Generation Time (seconds)
Iteration Count	1000	128	0.019
		256	0.040
		512	0.081
		1024	0.160
	2000	128	0.041
		256	0.080
		512	0.146
		1024	0.310
	4000	128	0.093
		256	0.166
		512	0.328
		1024	0.588

An analysis of the key generation algorithm shows that the 1024 bit key with 4000 iterations requires the most time to operate. Further analysis shows that the 1000 and 2000 iterations creates keys with a moderate time requirement. Increasing the number of iterations from 2000 to 4000 iterations impacts the time required by the system. Analysis shows that doubling the key size approximately doubles the time required for key generation provided the iteration count remain the same. Figure 4.5 shows the graph depicting the effect of key size and iterations on time requirements.

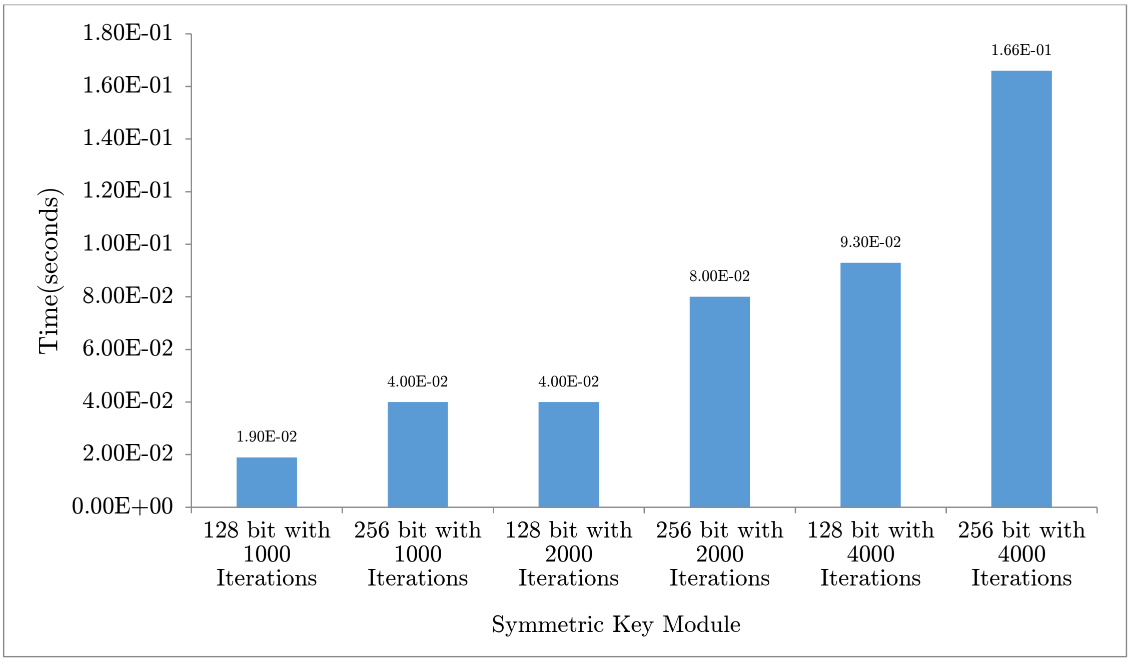

Figure 4.5. Graph showing time (seconds) taken by the various key sizes with a varying iteration count

The confidentiality module has been tested with two widely recognized encryption algorithms i.e. Rabbit stream cipher and AES. The rabbit stream cipher [140] has a single variant that requires a 128 bit key with a 64 bit initialization vector to run and requires only 7×10^{-6} seconds to run top down. Given in figure 4.6 is a graph showing the time taken by the individual encryption schemes and their variants.

The AES encryption [141] module is composed of two variants i.e. 128 bit and 256 bit. The 128 bit variant requires 3.6×10^{-6} seconds to run; while the 256 bit variant requires 5.1×10^{-6} seconds.

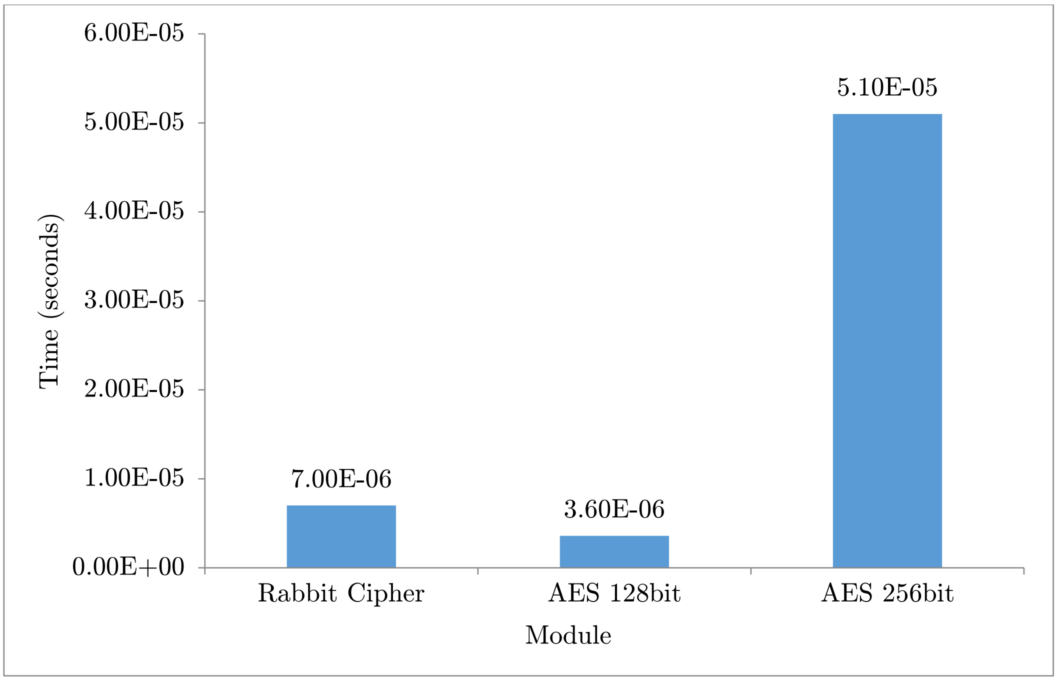

Figure 4.6. Graph showing time (seconds) taken by the Rabbit stream cipher and the AES variants

4.4.2 Scheme Analysis

The proposed scheme aims to provide a symmetric key for a multiparty environment. A single generated symmetric key is intended to be used by all parties of the group. The scheme offers forward and backward key secrecy because a new group ICMetric is generated whenever a participant joins or leaves the group. The joining or leaving of a group member triggers key revocation and a fresh symmetric key is generated for the group.

By incorporating secure channels and the ICMetric technology into the scheme discourages passive eavesdropping and man in the middle attack. To further strengthen the scheme noteworthy cryptographic elements have been incorporated like random salts and a large iteration count. Incorporating salts into the key generation algorithm defeats dictionary based attacks on the system and also ensures that a new key is generated every time even if the group membership remains the same. The iteration count is a crucial parameter of the PBKDF algorithm as an excessively large iteration count increases the time required to generate the key. Therefore a decision of how large an iteration count should depend on the application demand, system capabilities and time restrictions. PBKDF is an adaptable algorithm that allows designers to generate keys by specifying the required key length, associated secret input and the salt value at run time. Flexibility in design can increase the practicality of the system as varying key sizes can be generated based on application requirements.

Provision of strong authentication and key generation does not ensure a fully secure system therefore it has been studied in combination with two prominent confidentiality schemes AES and Rabbit.

4.5 COMPARATIVE ANALYSIS

Studying the PBKDF algorithm with the ICMetric technology is a novel concept that has not been explored previously. The simulation results of the proposed symmetric key scheme is compared to a healthcare sensing system [95] based on the ICMetric technology. The system is a one to one scheme that provides ICMetric based authentication and access control. The system also offers AES based encryption by using symmetric keys. Since the system was intended for one to one communication therefore the scheme is not constituent of a group ICMetric module. The scheme is initiated with the establishment of an ICMetric, followed by generation of the symmetric key. This symmetric key is then used to provide confidentiality services. The authors have simulated the scheme and their projected time consumption can be compared to the time requirements of the schemes proposed in this chapter. Authentication services and key exchange is carried out by a Secure Remote Password scheme and hence the authors have not provided a dedicated module for authentication. Table 4.3 below provides the time taken by this scheme and a rivalling scheme also based on ICMetric technology. As the contending scheme does not simulate all modules therefore absent details have been represented with a dash.

Table 4.3. A running time (seconds) comparison of the proposed symmetric key scheme with an ICMetric based one to one healthcare system

	Proposed scheme			ICMetric based one to one scheme		
Group ICMetric generation	1.5×10^{-3} sec			-		
Authentication	5.0×10^{-3} sec			-		
Symmetric key generation	1000 iterations			160 bit	256 bit	512 bit
	128 bit	256 bit	512 bit			
	1.9×10^{-2} sec	4.0×10^{-2} sec	8.1×10^{-2} sec	2.65×10^{-3} sec	3.6×10^{-3} sec	3.85×10^{-3} sec
AES 128	3.6×10^{-6} sec			3.1×10^{-6} sec		
AES 256	5.10×10^{-5} sec			-		
Rabbit encryption	7.0×10^{-6} sec			-		

4.6 SUMMARY

The ICMetric technology has been conceived to form a secure foundation upon which cryptographic schemes can be built. This chapter demonstrates that it is possible to generate symmetric keys that are based on ICMetric. The symmetric key generation scheme unifies the ICMetric technology and prominent cryptographic elements like PBKDF, hashing, cryptographic salts, AES and Rabbit. The ICMetric technology deters key theft by using the features of a sensor. The proposed scheme provides authentication by using the device

ICMetric. When authentication happens in a group setting an environment is created where only authenticated devices can communicate and share resources.

The key generation scheme uses the group ICMetric to generate a symmetric key using PBKDF. By incorporating PBKDF into the scheme design creates an adaptable method that allows creating cryptographic keys of variable size. The PBKDF also takes as parameter an iteration count. The iteration count increases the amount of computation required to generate a key thus making it difficult for an attacker to capture the key. By incorporating cryptographic salts throughout the designed system it deters dictionary based brute force attacks.

The proposed scheme has been simulated to ensure that keys are generated with minimum impact on time requirements. The symmetric key generation scheme has been tested with four prominent key sizes i.e. 128, 256, 512, 1024 bits with increasing PBKDF iterations. Simulation results show that the ICMetric technology can be coupled with the PBKDF algorithm with minimum impact on running time. The symmetric key generation scheme has been tested in AES 128, AES 256 and the Rabbit stream cipher. The proposed symmetric key generation scheme has been studied by comparing with a one to one scheme that uses the ICMetric technology. The comparison aims to prove that the ICMetric technology can be used to create a symmetric key for a group of devices. The proposed scheme competes closely with the rivalling scheme and delivers higher levels of security without compromise in the running time.

CHAPTER 5

ASYMMETRIC KEY BASED GROUP COMMUNICATION

Asymmetric key cryptography is an important tool for any cryptographer because of the advantages public key cryptography offers compared to symmetric key cryptography. Asymmetric key is termed as "asymmetric" owing to the way keying elements are held by the individual parties. Asymmetric keys are composed of two different keys i.e. a key which is made public and another which is private and held only by the owner. It is known that asymmetric key generation can be computationally intensive and its use may seem like an inconvenience but infact this form of keying possesses qualities which makes this an attractive alternative to symmetric keying. Asymmetric key cryptography is very different as compared to symmetric key cryptography because asymmetric keys are based on the computational intractability like the key generation may be based on unique large primes. Asymmetric key cryptography is a fundamental security ingredient for a wide range of cryptographic elements like digital signatures, Transport Layer Security (TLS), PGP, SSL etc. This chapter studies the generation of an ICMetric based asymmetric key using the widely accepted RSA algorithm. Coupling the

ICMetric technology with RSA is a novel concept that appreciates the security of the RSA algorithm and the target system. The chapter studies the individual building blocks of the asymmetric key generation scheme and then details of the algorithm are provided. The chapter also provides the simulation and evaluation results of the proposed scheme.

5.1 ASYMMETRIC KEY CRYPTOGRAPHY

The idea of using two keys instead of one was first explored in 1976 by Whitfield Diffie and Martin Hellman [142]. Their research formed the basis for a range of cryptographic services which we use today like, digital signatures and digital certificates.

Asymmetric key or public key cryptography is based on two keys i.e. a public key and a private key. The public key is widely disseminated and is not kept secret, while the private key is kept secret and steps should be taken to ensure its secrecy. In asymmetric key cryptography the public key and the private key possess a unique relationship such that it is mathematically infeasible to extract the private key if the public key is available. Although there are many individual applications of asymmetric key cryptography, its use can be best understood in the encryption decryption process.

When encrypting using asymmetric keys it must be highlighted that the public key is equally accessible to both an ally and aggressor. While the private key is kept secret and only the owner is aware of its contents. Figure 5.1 (a) shows the first scenario where Alice wishes to send a secret message to Bob. To do so Alice will use Bob’s public key to carry out the encryption. Bob will use his private key to carry out the decryption. In this setup an attacker is not able to carry out the decryption since only Bob’s private key can provide the correct plaintext. This scheme ensures that Alice and Bob are able to share a secret without having to share secret keys. Secondly, by using Bob's public key Alice can ensure that only Bob can decrypt the ciphertext.

Encryption in asymmetric key can also be carried out using a private key. Here the challenge is that Bob wants a guarantee that the message was actually sent by Alice and not by an impersonating adversary. In such a situation Alice will encrypt the message with her private key whereas Bob will use Alice's Public key to decrypt the message. The senders guarantee is provided since the private key is only in possession of the sender. Such an arrangement is not meant to offer secrecy as Alice's public key is also available to the adversary. Figure 5.1 (b) shows the second case where Alice sends a message to bob by using her own private key. A benefit of using this arrangement is that non-repudiation is implied if the private key has not been compromised. This type of scenario should be used with caution as the public key is accessible to all which means that the message can be decrypted even by an adversary.

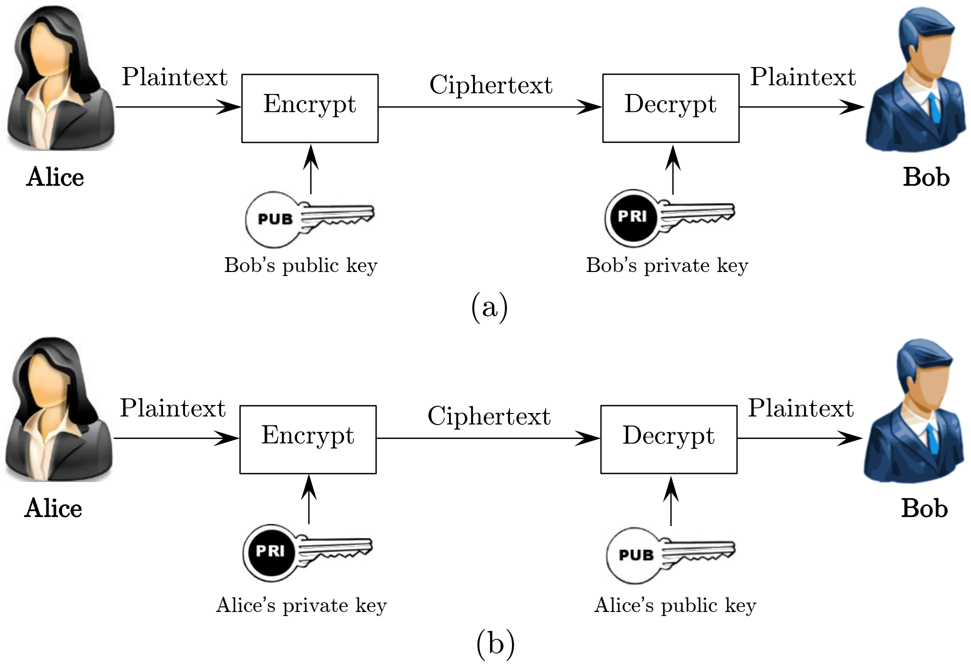

Figure 5.1. Encryption-decryption process using an asymmetric key (a) Encryption with a public key (b) Encryption with a private key

Definition 5.1. For an encryption and corresponding decryption transformations $\{E_e : e \in \mathcal{K}\}$ and $\{D_d : d \in \mathcal{K}\}$ where $\mathcal{K}$ is the key space, then under the asymmetric key cipher for each associated encryption/decryption pair (e, d), the key e is called the public key and the key d is called the secret key and for a given ciphertext c, it is computationally infeasible to find a message m such that $E_e(m) = c$ [22].

5.2 SCHEME PRIMITIVES

The asymmetric key generation scheme is based on a number of cryptographic primitives. Before providing a detailed description of how the asymmetric key is generated it is important to throw light on the scheme primitives.

5.2.1 Cryptographically Secure Pseudorandom Number Generator

Random number generators play an important role in many computer applications like cryptography, simulations, games, lottery, etc. The sole purpose of a random number generator is to produce random numbers. Randomness can have different meanings in different scenarios and applications. For example randomness has different meanings when simulating a coin tossing experiment, generating a random password, choosing a random back-off period for a nonresponsive server. Each of these tasks have their own requirements for a random number.

Although there are many different types of random number generators not all are suitable for use in cryptography [143]. Random number generators (RNG) can be broadly placed into three categories i.e. True RNG (TRNG), Pseudo RNG (PRNG) and Cryptographically Secure PRNG (CSPRNG). Figure 5.2 shows the classification of random number generators.

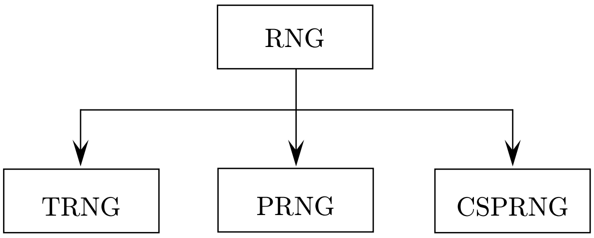

Figure 5.2. Classification of RNG into TRNG, PRNG and CSPRNG

TRNG use unpredictable sources to generate random data. Commonly used sources of random data are electrical resistor noise and oscillator phase noise. The problem [144] with a TRNG is that they produce data at a low data rate typically 20kbps. PRNG use an algorithm for generating a sequence of numbers that possess the same properties to those of truly random numbers. The problem with PRNG is that they lack uniformity of distribution and there maybe correlations between successive values. Often random number generators will fall short of some tests of true randomness and statistical analysis. Owing to this the use of a PRNG in a cryptosystem is discouraged.

The noncompliance of TRNG and PRNG to the field of cryptography has resulted in the creation of a PRNG which is suitable for use in cryptographic applications. CSPRNG are specialized PRNG which are designed to resist cryptographic attacks. A CSPRNG holds its security if it fulfils the following definition.

Definition 5.2. Given a sequence of k bits generated by a CSPRNG, it should be computationally infeasible to predict bit $k+1$ with confidence greater than $\frac{1}{2}$. Furthermore, if all or part of the internal state of the CSPRNG is revealed, it should not be possible to deduce the numbers previously generated [145].

There are just a few recognized CSPRNG algorithms for instance the CryptGenRandom function [146] is Microsoft's Cryptographic Application programming interface. Mac OS X and iOS devices use the Yarrow algorithm

[147] in their devices. Another popular CSPRNG algorithm is the Fortuna Algorithm [145] which was published in 2003. The OpenSSL cryptographic library uses CryptGenRandom function for the creation of random numbers.

5.2.2 Primality Testing

The purpose of a primality test is to determine if a number is prime or not. Many cryptographic algorithms like the RSA rely heavily on primality testing. Primality testing has been a focus of research for a long time because the aim has always been to improve the efficiency and correctness of the algorithm. Perhaps one of the earliest algorithms on primality testing is the Fermat's primality test which was based on Fermat's little theorem [148]. Another popular primality testing algorithm was the Solovay-Strassen [149] test published in 1977. This test was relegated in 1980 with the emergence of the Miller-Rabin [150] test. Miller-Rabin test offers better performance with at least the same correctness as its precursor. It still remains the most practical and widely used method of checking primality even though it is a probabilistic test. The OpenSSL library uses the Miller-Rabin test to check for primality. The function is provided in the BIGNUM multiprecision integer arithmetic library. The syntax [151] of the function is as follows:

```
#include <openssl/bn.h>

int BN_is_prime_ex(const BIGNUM *p, int nchecks, BN_CTX
*ctx, BN_GENCB *cb);
```

The basic object in this function is the BIGNUM which is a single large integer. This is considered as an opaque data type as the individual fields are not directly accessible. The function performs a Rabin-Miller probabilistic primality check with `nchecks` iterations. As the BIGNUM object can be fairly large therefore the creation and deletion of BIGNUM instances can be costly. To solve this problem the function uses BIGNUM context (`BN_CTX`) which is composed of a number of temporary BIGNUM linked lists and stacks that hold data temporarily. The last parameter required by the function is a callback which provides feedback on the progress which is specially required when the number of iterations is high. The function has a very low error probability which is less than 0.25^{nchecks} [152].

5.2.3 RSA Algorithm

The RSA cryptosystem is a result of research [153] by Ron Rivest, Adi Shamir and Leonard Adleman. The RSA algorithm still remains the most popular and successful algorithm for public key cryptography. Years of cryptanalysis of the RSA algorithm has not been able to prove or disprove its security. The RSA algorithm was the first algorithm which is based on the primality testing problem and the integer factorization problem.

Definition 5.3. The primality testing problem is a tractable problem and states that given a positive integer greater than 1, determine whether or not it is a prime [154].

Definition 5.4. The integer factorization problem states that given a large positive integer $n > 1$ find a factor $1 < p < n$ of n which satisfies the condition $n = pq$ where $1 < q < n$. The factors p and q should be large primes. The integer factorization problem remains unsolvable in polynomial time [154].

The RSA algorithm is composed of three individual components i.e. key generation, encryption and decryption. We limit ourselves to the key generation component as we modify this algorithm to function with the ICMetric technology. The RSA key generation algorithm works as follows:

1. Generate two large random primes p and q.
2. Compute the RSA modulus $N = p * q$.
3. If φ is the Euler's totient function then compute the private exponent.

$$\begin{aligned} \varphi\,(N) &= \varphi(p)\,\varphi(q) \\ &= (p-1)(q-1) \end{aligned} \tag{12}$$

4. Choose an integer e such that $1 \leq e \leq \varphi(N)$ i.e. e and $\varphi(N)$ are coprimes.
5. Compute the public exponent d.

$$d \equiv e^{-1}(mod(N)) \tag{13}$$

The public key exponent d is released whereas the private component φ is kept secret.

The RSA algorithm uses distinct large primes to compute the modulus. At this point it is natural to question what would happen if there are no more unique primes. Euclid [155] answers this question in his theorem where he proposes that there are infinite many primes number. The theorem has been studied and proved correct in other researches [156][157]. For a system designer the concern is to generate strong enough primes quickly. Obviously, generating large unique primes will take more time compared to a smaller arbitrary prime.

5.3 ASYMMETRIC KEY GENERATION

The ICMetric based asymmetric key generation scheme uses hashing, CSPRNG, primality testing and the RSA algorithm. Once the group ICMetric is generated it is hashed and used as a seed to create two CSPRNG. The two primes are tested for primality and then used for creating public and private key pairs. Given below are the five steps of the proposed asymmetric key generation scheme based on ICMetric.

1. Generate the group ICMetric icm_g.
2. Hash the group ICMetric icm_g to create a seed of 128 bit length.
3. Create two cryptographically secure pseudo random numbers by using seed in a CSPRNG.

4. Test for primality and repeat step 3 until two large primes p and q are obtained.
5. Process the primes p and q in the RSA algorithm to generate the public and private key pairs.

Given in figure 5.3 is a flow diagram which shows the sequential flow of the proposed asymmetric key generation scheme based on ICMetric.

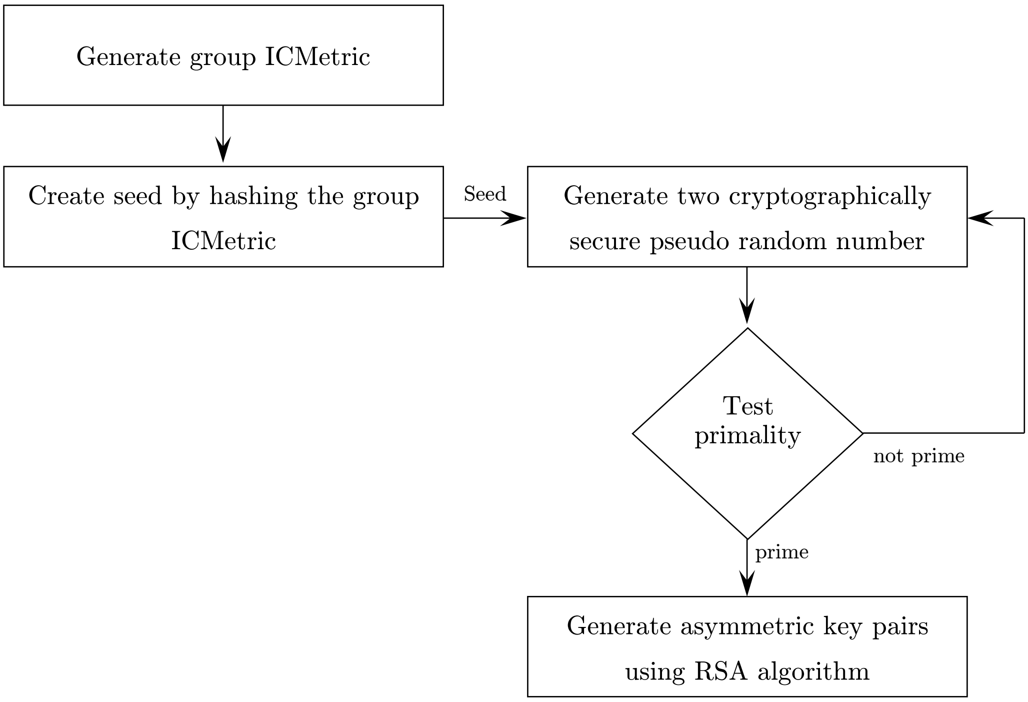
Generate group ICMetric
Create seed by hashing the group ICMetric
Seed
Generate two cryptographically secure pseudo random number
Test primality
not prime
prime
Generate asymmetric key pairs using RSA algorithm

Figure 5.3. The ICMetric based asymmetric key generation scheme

5.4 IMPLEMENTATION AND OUTCOMES

The proposed ICMetric based asymmetric key generation scheme has been implemented and tested on Intel Core i5 3.4GHz processor computer with 6GB RAM. The scheme has been programmed using Bloodshed Dev-C while the cryptographic functionalities are provided by the OpenSSL cryptographic library [139].

5.4.1 Outcomes

The proposed system is tested by generating common cryptographic key sizes i.e. 128, 512, 1024, 2048 bit. Even though the 128 bit key is a size which is not favored owing to fear of being captured, its use is justified as keys are periodically regenerated to provide key freshness in the group. Given in figure 5.4 is the graph giving key generation times for various key sizes.

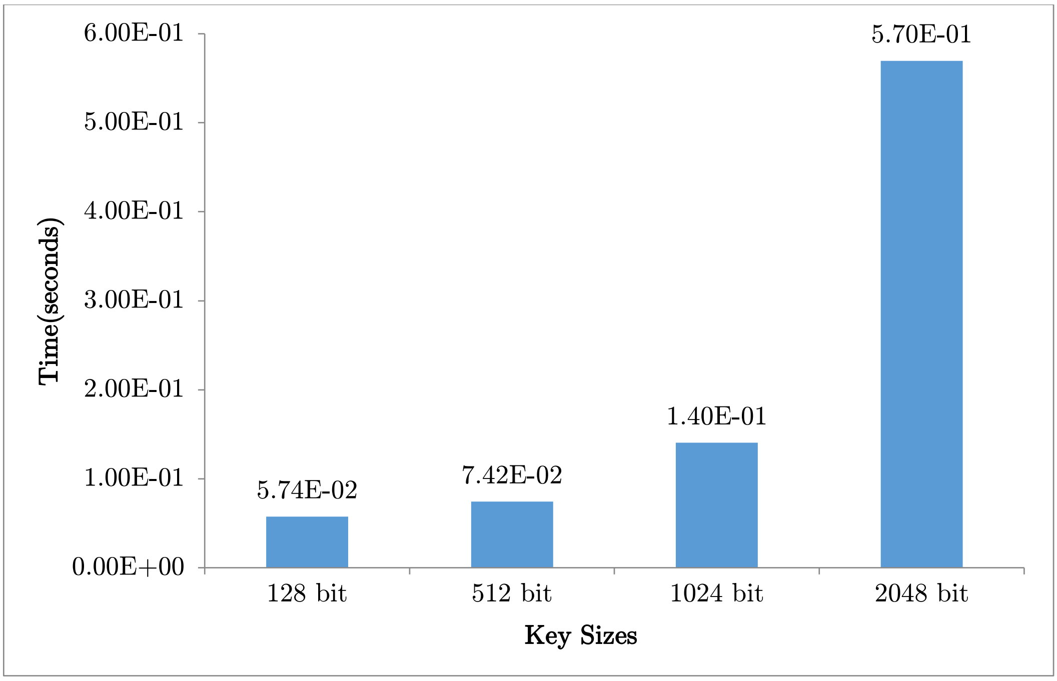

Figure 5.4. Graph showing time (seconds) taken by the ICMetric based RSA scheme to generate keys of size 128, 512, 1024, 2048 bit

5.4.2 Scheme Analysis

The proposed scheme presents an asymmetric key generation scheme that is based on ICMetric. The scheme offers forward and backward secrecy as a new key pair is created whenever a new group ICMetric is generated. In the scheme it is assumed that all communications take place via secure channels which helps in deterring man in the middle and eavesdropping attacks on the system.

The scheme is initiated once a group ICMetric is assembled. The group ICMetric is hashed to create a seed which is used for generating two large primes required for RSA algorithm. By using a seed for prime generation the scheme ensures diversity and unpredictability of primes. The resulting primes are tested for primality and then used in the RSA key generation algorithm. The proposed scheme can be tailored to any key size that is supported by RSA. Thus the presented scheme increases system practicality as varying key sizes can be generated based on application requirements.

5.5 COMPARATIVE ANALYSIS

This thesis is an effort to secure group communications using the ICMetric technology. Previously the ICMetric technology has not been coupled with the RSA algorithm. An advantage of coupling the ICMetric technology with RSA is that it allows private key encryption with public key decryption. This setup provides authentication guarantees to the receiver. This form of authenticated message exchange is not provided by PUF. Thus the proposed asymmetric key generation scheme can only be compared with recent implementations of the RSA algorithm. The proposed scheme can be compared with two schemes that utilize the RSA key generation algorithm. The first research by Vijayalakshmi et al. [158] is a multiparty key agreement protocol that studies identity based authentication with Elliptic Curve Cryptography (ECC) and RSA. The authors study the RSA algorithm and experiment with two keys i.e. 128 bit and 1024 bits. The authors have not experimented with any other key sizes which is a shortcoming of the scheme. The 128 bit key is quite dated, as key sizes have increased and the 128 bit key is often considered weak. The work has been simulated on an Intel Pentium Dual Core 2.2GHz processor with 2GB RAM.

The second scheme by Dongjiang et al. [159] studies the RSA algorithm for public key cryptography. The authors have proposed methods to improve the efficiency of the RSA by incorporating a pre-screening algorithm. The pre-screening algorithm streamlines the prime number generation module to improve performance of the overall RSA key generation. The authors have experimented with 1024 bits and 2048 bits to study the improved RSA algorithm. The authors have not given the specifications of system on which the simulation was conducted.

The proposed asymmetric key scheme has been simulated with four key sizes i.e. 128, 512, 1024, 2048 bit. Table 5.1 provides a running time comparison of the ICMetric based asymmetric key generation scheme and two RSA based key generation schemes. As the other schemes do not simulate all the key sizes therefore absent details have been represented with a dash.

Table 5.1. A running time (seconds) comparison of the proposed scheme with Vijayalakshmi et al. scheme and Dongjiang et al. scheme

	Proposed scheme	Vijayalakshmi et al. scheme	Dongjiang et al. scheme
128 bit	5.74×10^{-2} sec	1.334×10^{-1} sec	-
512 bit	7.42×10^{-2} sec	-	-
1024 bit	1.404×10^{-1} sec	4.001×10^{-1} sec	8.8×10^{-2} sec
2048 bit	5.696×10^{-1} sec	-	1.84×10^{-1} sec

By analysing the tabulated running times it can be concluded that the ICMetric technology offers a secure method of supporting key generation with minimum impact on running time. The proposed scheme does not outperform rivalling schemes but the performance difference is minute which is why it should not have a significant impact on the practicality of the resulting system. The extra running time can be justified when one considers the benefits ICMetric technology offers compared to conventional cryptographic systems.

5.6 SUMMARY

The ICMetric technology can be used for the provision of cryptographic services. Many modern cryptographic systems are based on using asymmetric keys to provide security. This chapter studies the ICMetric technology as a basis for asymmetric key generation. Asymmetric key cryptography uses a combination of public and private keys for the provision of cryptographic services. Asymmetric keys have become a popular keying mechanisms because of the way the keys are held by the owner and the public. A security algorithm that has been able to resist a wide range of attacks is the RSA algorithm. The RSA algorithm is the most widely used asymmetric key cryptosystem. The popularity of the RSA technology encourages the creation of an algorithm that combines both the ICMetric technology and the RSA algorithm.

The strength of the RSA algorithm lies in the generation of two large primes which are used for computing a modulus. Even though the security of the algorithm lies in the largeness of the primes, one cannot deny the fact that keeping the keys secret once they are generated is an essential part of ensuring secrecy of the cryptosystem. If the private key is captured then an RSA based system is fully exposed. The ICMetric technology enhances the security of RSA through the use of key theft deterrence.

This chapter has studied the unification of the ICMetric technology and RSA for security in a multiparty environment. The proposed algorithm provides a method of generating a public private key pair based on the ICMetric technology. The algorithm uses the group ICMetric, hashing, cryptographically secure pseudo random number generators, primality testing and the RSA algorithm to create an asymmetric key pair for the group. Efforts have been made to ensure that the ICMetric technology works seamlessly with the RSA algorithm so that security and practicality of the resulting system is enhanced. Owing to the sound design of the RSA, the algorithm has not been modified to make it adapt to the ICMetric technology. An advantage of coupling the ICMetric technology with RSA is that the recipient of an encrypted message can authenticate the sender. This form of authentication is rooted on the key theft deterrent quality of the ICMetric technology.

The proposed scheme has been simulated to ensure that keys are generated with minimum impact on time requirements. The asymmetric key generation scheme has been tested with four prominent key sizes i.e. 128, 512, 1024, 2048 bits. Simulation results show that the ICMetric technology can be coupled with the RSA algorithm with minimum impact on running time. The simulation results have been compared with two recent researches. Analysis shows that the proposed scheme requires slight more time to operate but also offers more security owing to which its use can be justified in multiparty environments.

CHAPTER 6

SYSTEM ANALYSIS

Designing provably secure cryptographic schemes is a difficult task because security is often achieved at the cost of reduced efficiency. The task becomes even more complex when it is both difficult to predict and simulate the amount of resources available to an adversary. Cryptographic algorithms are a sequence of activities that follow a defined order. Hence cryptographic workflow is a necessary part of ensuring that the system functions as intended. Therefore research often uses mathematical intractability to prove the security of an algorithm. This chapter presents a security analysis of the ICMetric technology and its interaction with cryptographic elements used in various modules throughout the thesis.

This chapter proves that the ICMetric technology supports and enhances the security of primitives and the proposed system. Unifying multiple systems does not guarantee a sufficiently secure system as adversaries may attempt to exploit unpredictable weakness in the schemes. This chapter proves that the schemes presented in this thesis do not contradict each other and that the properties of the ICMetric technology are also preserved. This chapter studies the proposed schemes in the standard model and uses security proofs to shows that the ICMetric technology works hand in hand with cryptographic schemes and primitives for the creation of a secure multiparty communication framework.

6.1 PROVING SECURITY

The simulation of a security algorithm can give insight into how the individual algorithmic elements interact and the time required by the algorithm. A guarantee limited to these two factors is not sufficient to prove the security of a cryptographic algorithm. A cryptosystem can be subjected to security analysis by following one of two standard models i.e. the Random Oracle Model (ROM) or the Standard Model. When testing cryptographic schemes it is often difficult to predict how an adversary will behave and what resources it has available. This serves as a motivation to model stronger adversaries so that highly secure cryptosystems can be constructed.

The ROM [160][161] gives heuristic confidence in the design soundness of a cryptographic scheme. The design soundness of an algorithm shows that the individual building blocks work together but this does not shows how an algorithm would behave in the real world. Owing to the lack of ability to model real world scenarios, theoretical and practical research on cryptography is often based on the Standard model [162][163].

Proving the security of a cryptosystem is a complex task especially when it is not always obvious how much resources an adversary will have to attack the system. Therefore research often uses mathematical intractability to prove the security of an algorithm. The security of popular algorithms like the RSA is still based on the intractability of the integer factorization problem [164]. The security of a cryptographic scheme can be proved under the standard model by deliberately modelling the presence of an adversary which is able to break down the scheme and its primitives. All legitimate parties are expected to behave according to the defined algorithm. Thus to an onlooker the legitimate parties follow an algorithm exactly as they would in real world. An advantage of the standard model is that it allows the replication of interactions following the precise mathematical procedures and sequence as defined in the algorithm.

6.2 FORMAL PROOFS IN THE STANDARD MODEL

Provable security was first studied in 1989 by Shafi Goldwasser and Yael Tauman [165]. Provable security is the process of justifying the security of a cryptographic scheme in the presence of an adversary. To prove security, well studied atomic primitives are chosen that form building blocks of the scheme. It is then demonstrated that the security scheme “works” through reductions. The reductions show that the only way to break the scheme is by breaking the atomic primitive. A benefit of carrying out such an exercise is that there is no need to perform a cryptanalysis of the scheme as it is based on a provably secure atomic primitive. If the latter is secure then it can be inferred that the underlying scheme is also secure [166].

A cryptographic proof under the standard model has two essential elements i.e. an adversary model and a security proof.

- Adversary model - The adversary model is a formal definition of the adversary. The model defines if the adversary is passive or active while carrying out an attack.
- Security proof - The security is composed of what elements an adversary has access to. The security proof also defines the starting point of interactions and when an adversary has broken the cryptosystem. The security proof is composed of a challenger who challenges the adversary to break the cryptosystem when it is given access to information and elements. Outcomes of the security proof are used to determine the extent of damage if certain pieces of information are exposed to the adversary.

6.2.1 ICMetric Security

When working with the ICMetric technology it is important to prove that it offers higher levels of security compared to conventional cryptography. The ICMetric of a device cannot be transmitted which leaves just one possibility of capturing this unique device identity i.e. recreating the ICMetric if an attacker has access to a subset of device features. If an attacker is able to forge/ recreate *all* the features of a device then generating the ICMetric of the device is an effortless task. In this section a security proof is provided which demonstrates the security of the ICMetric technology.

Suppose $\mathcal{C}$ is a finite set of credentials (explicit and implicit features) required for the generation of a device ICMetric. It is known that each device has its own credentials $c \in \mathcal{C}$.

The ICMetric generation is a deterministic algorithm. When the correct set of credentials c are presented to the ICMetric generation algorithm Θ then it returns a single ICMetric which can uniquely identify that system.

6.2.1.1 ICMetric Producibility

Based on observation of device behaviour an adversary may attempt to predict the feature of device to forge a device ICMetric. The notion of producibility requires that it should be difficult for an adversary to produce the device ICMetric such that he has access to some device features. All interactions happen between the challenger Ch and an adversary $\mathcal{A}$.

Security Proof

Let Θ be an ICMetric generation algorithm that takes as input a set of explicit and implicit features $c \in \mathcal{C}$. The algorithm produces (within polynomial time) a unique device ICMetric ICM_d.

Setup Phase. The challenger Ch generates its credentials $c \in \mathcal{C}$ under a standard stimulus.

Challenge Phase. The challenger Ch sends the subset $c_1 \in c$ to the adversary $\mathcal{A}$ along with the ICMetric generation oracle. The adversary $\mathcal{A}$ receives c_1 and the ICMetric oracle. The adversary uses illicit software and hardware to produce c_{ad} such that it has knowledge of c_1. The adversary $\mathcal{A}$ sends the set c_{ad} to the challenger Ch.

Outcome Phase. The challenger Ch receives c_{ad}. The challenger now provides outcome O such that $O \in \{0,1\}$. The determination of O is as follows:

$$\text{If } c_{ad} \subseteq c \text{ then output } O = 0$$

$$\text{If } c_{ad} \subset c \text{ then output } O = 1$$

Therefore if $O = 1$ the adversary $\mathcal{A}$ has been unsuccessful in producing the credentials c thus the challenger Ch has won.

As the ICMetric generation is based on a large number of explicit and implicit features therefore it is difficult for an attacker to forge an ICMetric. By using illicit hardware and software tools an adversary may capture some features, but cannot capture the entire feature set as the ICMetric is based on explicit and implicit features. Many of the features are physically unclonable which will deter feature theft.

6.2.1.2 ICMetric Preimage Hash Resistance

Hashing is one of the most widely used cryptographic tool. It is important to show that by using the ICMetric technology with hashing does not violate the properties of both the ICMetric technology and hash functions. Perhaps the greatest concern when using hashing with ICMetric is the possibility of illicitly extracting a device ICMetric from its hash image. Formally called the preimage hash resistance property, it dictates that given a hash function $h : \mathcal{X} \to \mathcal{Y}$ find $x \in \mathcal{X}$ where $y \in \mathcal{Y}$ and $h(x) = y$. Interactions take place between a challenger Ch and an adversary $\mathcal{A}$.

Security Proof

Let Θ be an ICMetric generation algorithm that takes as input a set of explicit and implicit features $c \in \mathcal{C}$. A publically available hash function h that is known both to the challenger Ch and the adversary $\mathcal{A}$.

Setup Phase. The challenger Ch communicates to the adversary $\mathcal{A}$ the hash image y obtained by hashing its own ICMetric ICM_d.

$$y = h(ICM_d) \qquad (14)$$

Challenge Phase. The adversary $\mathcal{A}$ generates an ICMetric ICM_d' based on a set of features. The adversary then computes $h(ICM_d') = y'$. The obtained hash image y' is communicated to the challenger Ch.

Outcome Phase. The challenger Ch receives y' and provides an outcome O. The outcome $O \in \{0,1\}$ is determined as follows:

If $y' \equiv y$ then output $O = 0$

If $y' \not\equiv y$ then output $O = 1$

Therefore if $O = 1$ the adversary $\mathcal{A}$ has been unsuccessful in extracting the ICMetric of a device when it had access to the hash image of the ICMetric thus the challenger Ch has won.

If the output of an n-bit hash function has been provided then producing a preimage requires approximately 2^n operations which makes the task computationally infeasible [167].

6.2.2 Device Authentication

Using ICMetric as a method of facilitating authentication requires that systems are able to authenticate each other without transmitting their own ICMetric in its pure form. Since the ICMetric of a device cannot be transmitted then a challenge is to ensure that genuine entities gain access to the group. The purpose of this security proof is to demonstrate that it is not possible for an adversary to produce or assume a fake ICMetric and then act like a genuine entity. The interactions take place between a challenger Ch and an adversary $\mathcal{A}$ in the presence of a key generation centre kg.

Security Proof

Let Θ be an ICMetric generation algorithm that takes as input a set of explicit and implicit features $c \in \mathcal{C}$. A publically available hash function h that is known both to the challenger Ch and the adversary $\mathcal{A}$.

Setup Phase. The challenger Ch communicates a hash of his ICMetric ICM_d to the key generation centre kg as follows.

$$a = h(ICM_d) \tag{15}$$

The challenger Ch is issued a temporary salt by the key generation centre kg. The challenger Ch receives the temporary salt s and communicates it to adversary $\mathcal{A}$.

Challenge Phase. The adversary $\mathcal{A}$ is allowed to communicate with the kg for authentication. The adversary $\mathcal{A}$ computes its version of equation 10 to produce a' by producing an ICMetric ICM'_d from a set of features.

$$a' = h(ICM'_d) \tag{16}$$

The computed a' is communicated to the key generation centre kg.

Outcome Phase. The key generation centre kg will provide outcome as follows:

The key generation centre kg will independently compute the following based on inputs from the challenger Ch and the adversary $\mathcal{A}$:

$$a_2 = h(a + s)$$

$$a_3 = h(a' + s)$$

The outcome $O \in \{0,1\}$ is determined as follows:

$$\text{If } a_2 \equiv a_3 \text{ then output } O = 0$$

$$\text{If } a_2 \not\equiv a_3 \text{ then output } O = 1$$

Therefore if $O = 1$ the adversary $\mathcal{A}$ has been unsuccessful in getting itself authenticated in place of the challenger Ch in which case the challenger Ch has won.

ICMetric based authentication is based on the secrecy of the ICMetric of a device. Since the ICMetric of a device is kept secret therefore any computation involving the ICMetric of the device makes the computation indeterministic.

6.2.3 Key Freshness

Using the ICMetric as a basis for key generation in a multiparty environment requires that the keys are kept secret and that perfect forward and backward secrecy is ensured. These cryptographic properties safeguard the group from misuse of keys by entities that are part of or have been part of the group.

6.2.3.1 Perfect Forward Secrecy

Perfect forward secrecy in a group environment requires that once a member leaves a group he does not have access to ongoing communications and keys of the group produced thereafter. The interactions take place between a challenger Ch and a group member i where $i \in \mathbb{N}$ (later termed as an adversary $\mathcal{A}$).

Security Proof

Let Θ be an ICMetric generation algorithm. An ICMetric key generation algorithm δ is used to generate a symmetric key or asymmetric key for n participants in the group.

Setup Phase. The challenger Ch, at random selects two encrypted messages $Enc_k(m_0), Enc_k(m_1)$, where k represents the group key and sends them to the group member i. On receiving the encrypted messages sent by the challenger Ch, the group member i decrypts the messages using the same group key k. The group member i terminates his membership following which he is no longer a member of the group and is classified as adversary $\mathcal{A}$. The group key k is revoked and a new key k' is generated.

Challenge Phase. The adversary $\mathcal{A}$ is allowed to generate two messages m_0', m_1' and send them to the challenger Ch. The selection of the messages is made such that:

$$m_0',\ m_1' \in \{m_0, m_1\} \text{ and } m_0' \neq m_1'$$

The challenger Ch tosses a fair coin $b \leftarrow \{0,1\}$ and encrypts the message m_b' with the key k' and send it to the adversary $\mathcal{A}$.

$$Enc_{k'}(m_b')$$

Outcome Phase. The adversary $\mathcal{A}$ receives the encrypted message. By analysing this newly generated message against the previously stored messages m_0, m_1, the adversary $\mathcal{A}$ has to guess and output bit b. The adversary $\mathcal{A}$ wins if the bit b is guessed correctly otherwise the scheme provides perfect forward secrecy.

Therefore, the probability that the adversary $\mathcal{A}$ wins is $\frac{1}{2}$.

6.2.3.2 Perfect Backward Secrecy

Perfect backward secrecy in a group environment requires that once a party joins a group it does not have access to communications and keys utilized prior to its introduction into the group. The interactions happen between a challenger Ch and a group member i where $i \in \mathbb{N}$.

Security Proof

Let Θ be an ICMetric generation algorithm. An ICMetric key generation algorithm δ is used to generate a symmetric key or asymmetric key for n participants in the group.

Setup Phase. There are n participants in the group who share a common group key k_j where $j \in \mathbb{N}$. The participants encrypt and decrypt messages using the key k_j. Challenger Ch maintains a history of all past encrypted messages $Enc_{k_j}(m)$. A dishonest group member i (hereafter called the adversary $\mathcal{A}$) joins the group. The group key k_j is revoked and a new key k_{j+1} is created.

Phase I. The adversary $\mathcal{A}$ encrypts and decrypts messages in the group using the group key k_{j+1}.

Challenge Phase. The challenger Ch selects an encrypted message $Enc_{k_j}(m)$ from his history and sends it to the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ executes phase I again.

Outcome Phase. The adversary $\mathcal{A}$ wins if he is able to output the decrypted message m or guess the associated plain text. Thus the adversary must correctly output $Dec_{k_j}(m)$ to show that the system is penetrated.

When the adversary joins the group the previous keys are revoked and new keys are issued. This feature ensures backward secrecy and prevents access to previous communications of the group.

6.2.4 Compromised Client

A concern when communicating in group communications is the presence of compromised or dishonest participants. In these circumstances there is an issue that the secrecy of the ICMetric and the cryptographic keys will not be maintained.

Security Proof

Let Θ be an ICMetric generation algorithm. An ICMetric key generation algorithm δ is used to generate a symmetric key or asymmetric key for n participants in the group.

Setup Phase. There are n participants in the group who share a common group key k_j where $j \in \mathbb{N}$. A group member i has been unknowingly compromised owing to which his communications are being captured by an active adversary $\mathcal{A}$.

Phase I. The adversary $\mathcal{A}$ encrypts and decrypts messages in the group using current the group key k_j.

Challenge Phase. The challenger Ch sends an encrypted challenge-response test to group member i and the adversary $\mathcal{A}$. As the adversary and the group member have access to the same key therefore they will reply with a response message.

Outcome Phase. The group member i responds to the challenge with a message m_b. While the adversary $\mathcal{A}$ responds with message $m_{b'}$ The adversary $\mathcal{A}$ wins if the responses obtained are identical. Thus the outcome $O \in \{0,1\}$ is determined as follows:

$$\text{If } m_b \equiv m_{b'} \text{ then output } O = 0$$

$$\text{If } m_b \not\equiv m_{b'} \text{ then output } O = 1$$

Therefore if $O = 1$ the adversary $\mathcal{A}$ has been unsuccessful in getting access to the group and its communications in which case the challenger Ch has won.

ICMetric based authentication is based on the secrecy of the ICMetric of a device. If a group member is compromised then the device ICMetric and group ICMetric can be exposed which can result in authentication abuse and cryptographic key theft.

6.3 SUMMARY

Designing a cryptosystem by incorporating popular cryptographic primitives does not provide sufficient security guarantees. Although the stability of the individual primitives is often well understood the same cannot be said about their interactions with other primitives. The ICMetric technology provides a reliable root of trust upon which a cryptographic scheme can be built. When using the ICMetric technology with security primitives it is important to ensure that the design does not possess any technical flaw which could be exploited by an adversary to bring down the system. The various schemes presented in earlier chapters are based on strong atomic primitives. No doubt some schemes would continue functioning even if the atomic primitive was removed, but it must be highlighted that the primitives have been incorporated to provide higher level of security.

This chapter uses the standard model to prove that interactions between the scheme elements do not violate the cryptographic and ICMetric technology goals. There are many methods of proving security but often they do not model the physical world correctly. Security proofs have been designed in the standard model because this method accurately models scheme elements in the presence of an adversary.

Security proofs show that the ICMetric cannot be produced by an adversary even if a subset of features are disclosed to it. As the ICMetric of a device is never transmitted in its original form therefore a security proof demonstrates that it is not possible to capture an ICMetric if a hash is provided. After establishing security of the ICMetric technology the chapter shows that an adversary cannot exploit the authentication algorithm to gain unlawful admission to the group. This security proof also demonstrates that the ICMetric technology can be uniquely used to identify a device.

The aim of this thesis was to establish that the ICMetric technology can be used to generate keys for the group environment. Generating keys for a group requires that key freshness is maintained along with perfect forward and backward secrecy. As the schemes have been designed to function in the multiparty environment therefore it is important to ensure that the keys are kept secret when a group member joins or leaves the group.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

Cryptographic schemes and algorithms are tools that are used to protect users, devices and data. Users are becoming increasingly dependent on sharing data and they assume that their devices and data are safe from adversaries. Therefore it is the responsibility of the designers and manufacturers to ensure that this is the case. At present some devices for environments like the IoT are being designed and manufactured with little or no security provisions. This thesis has studied the security of devices communicating in a group environment such as IoT. The thesis studies incorporating the ICMetric technology as a key theft deterrent and a basis for a range of cryptographic services. This chapter provides a summary of contributions that were aimed at revolutionizing the security of devices in the multiparty environment.

7.1 PRACTICAL IMPLEMENTATION

In this thesis the security of devices in the multiparty environment has been studied. A testbed of wearable Shimmer sensors has been used for studying features suitable for ICMetric generation. The cryptographic services have been simulated in C-language using the OpenSSL cryptographic library.

The proposed schemes could be practically implemented on many network capable systems like wearable devices, smartphones, tablets and laptops. Some portable devices may not be able to support cryptographic services as they often lack power and processing capabilities required by cryptographic primitives. A necessary requirement for using the proposed ICMetric generation is that the target device must be embedded with MEMS based inertial sensors.

7.2 SUMMARY OF CONTRIBUTIONS

The security of cryptographic schemes lies in keeping the key secret while the scheme is made public. The Kerckhoff's principle states that *"only secrecy of the key provides security"*. This implies that cryptographic key theft can be the breaking point of any security system. The cryptographic key is a necessary piece of information which is able to ensure the security of the system. Typically a cryptographic key is a hexadecimal block of data that can range from 80 bit to beyond 256 bit in size. The cryptographic keys are never memorized owing to their size and data type which is why they are often stored on the system so they can be retrieved when needed [31]. If the keys are stored in a retrieval system then they can be attacked by an adversary through many methods. The many methods of attacks on various forms of computation devices are discussed in chapter 2. The chapter also studies cryptographic key theft as a fundamental problem in cryptography. This problem remains largely unexplored and can be referred to as an Achilles heel in a cryptographic system. Increasing key size can only reduce brute force attacks on keys [8] but does not entirely eliminate the possibility of key theft. To create stronger cryptosystems cryptographers now consider alternate roots of trust like PUF's, machine fingerprinting.

This thesis explores the use of ICMetric technology as a key theft deterrent and as a basis for security provision in a group setting. The ICMetric technology allows the creation of a device identification by using the features of the device. This device identification is then used for authentication and also to generate symmetric and asymmetric keys for the group. The ICMetric technology functions as a key theft deterrent because the key is only generated when required and discarded thereafter. This means that for an adversary aiming to capture a key there is nothing to attack or capture.

Analysis of commonly used features for device fingerprinting shows that many device fingerprinting based systems offer a false guarantee of security as the features they use can be easily captured and spoofed. In chapter 3 the Shimmer sensor has been used to study features that are suitable for ICMetric generation. Hence the first contribution of this thesis is an in-depth study of explicit and implicit features of a device which can used for ICMetric generation. The thesis proposes that the ICMetric of a device can be based on the bias in MEMS accelerometer, gyroscope and strain gauge sensors. Statistical analysis of the sensors readings showed that each sensor possesses a unique bias which can be used for ICMetric generation. The chapter also presents a study on explicit features which are relatively easy to generate but can uniquely identify a device.

In Chapter 3 the concept of the group ICMetric has been presented. The group ICMetric is an identity of the group and is used to identify groups of communicating devices. Hence the second contribution of this thesis is a group ICMetric generation scheme that is based on Shamir Secret Sharing. The scheme allows a group of devices to generate a group identity which can be used for the provision of security services like key generation.

Chapter 4 of the thesis demonstrates the fact that the ICMetric technology can be used to create cryptographic keys. The third contribution of this thesis is a symmetric key generation scheme that is based on PBKDF [134]. The PBKDF algorithm is based on an iteration count that can be increased or decreased based on the complexity of the system. Coupling the ICMetric technology and the PBKDF allows the creation of a strong symmetric key for the group environment. The proposed algorithm can be adapted to various key sizes by modification of the key size parameter taken by the function.

The thesis also establishes that the ICMetric technology can be used to create an asymmetric key. Chapter 5 studies public key cryptography in relation with the ICMetric technology. Asymmetric key cryptography is often favoured over symmetric keys due to the unique way they execute but are more resource intensive compared to symmetric keys. Hence the fourth contribution of the research is an ICMetric based asymmetric key generation scheme that creates keys with minimum overhead. The RSA algorithm is one of the most widely used asymmetric key generation algorithms because of its non-tractable algorithm. By using the RSA with ICMetric the resulting scheme creates an asymmetric key without compromising system security or the security of the algorithms involved.

Founding the design of any scheme on popular primitives does not guarantee a secure system. Therefore it is important to test a scheme for conformance to high levels of security. Hence any scheme that is proposed should be resilient to attacks on the individual components and their interactions. Chapter 6 studies the proposed ICMetric and key generation schemes through the standard model. By using the standard model interactions take place such that certain pieces of information are shared with the adversary so that the system can be attacked. The standard model is used the study the extent of damage if crucial pieces of information are exposed to an adversary.

7.3 FUTURE DIRECTIONS

It can be said that internal attacks are difficult to prevent, as trusted parties abuse their access rights to attack a system. By incorporating the latest features into an ICMetric, better intrusion detection systems can be designed.

The thesis has demonstrated the creation of an ICMetric by using various explicit and implicit device features. By no means are the presented features an exhaustive list. Therefore research should be aimed at the discovery of more features which can strengthen the ICMetric of a device. Incorporating network, operating system and web browser characteristics into the ICMetric technology can be a novel way of detecting both internal and external intrusions. The ICMetric technology can also be studies as a physically unclonable function thus promoting its use in areas outside cryptography.

As mentioned in chapter three the MEMS sensors are susceptible to varying behaviour due to environmental factors, fatigue, stress and aging [126][127][128]. The effect of these factors needs to be studied for reducing impact on the ICMetric generation process. Incorporation of correction codes into the ICMetric generation process can be used to significantly lower the impact of external factors on MEMS sensors.

The ICMetric technology does not possess self-correcting properties. Thus when a device ICMetric is being generated, erroneous bits due to environmental variations [168] will influence the resulting ICMetric. To correct this a self-correcting code will ensure robustness and stability of both the ICMetric and the resulting cryptographic keys.

The group ICMetric uniquely identifies a group of communicating devices. Efficient and secure methods are required that allow the generation of a group ICMetric without compromising the security of the ICMetric technology. Multiparty computations provide a method for communicating parties to generate a function over secret inputs. The concept can be incorporated into the group ICMetric generation process. Multiparty computations are often based on very simple arithmetic algorithms that use intractability for secrecy. This concept can be very effective for computing the group ICMetric.

7.3 FUTURE DIRECTIONS

Elliptic Curve Cryptography (ECC) is a public key approach, which has attracted a lot of attention owing to the advantages it offers over RSA based cryptosystems [169]. ECC can be used in portable and wireless devices because it carries out key generation, signatures, encryption and decryption better than RSA. Thus the ICMetric technology can be studied with ECC for improved performance and security.

For improved security and practicality, the ICMetric technology can be implemented as a System-on-Chip. Previous experiments [170] on hardware implementations of the ICMetric technology did not include programmable single board computing devices. Further experiments can include implementing the ICMetric technology using ARM microprocessor, PIC microcontroller, Atmel microcontrollers etc. Integration of various MEMS sensors and hardware implementation of ICMetric can be used to detect physical attacks on computation systems.

Cryptocurrencies and electronic payment systems are rapidly becoming part of daily life. The ICMetric technology has not been studied in block chains and Bitcoins. Bitcoins are not anonymous [171] by design which can present a weakness. This implies that people can find out who is using bitcoin and how they are spending them. If the ICMetric technology is integrated into bitcoin then it can provide both anonymity and promote secrecy of transactions.

This thesis has demonstrated that integrating the ICMetric technology into a wearable device will improve the safety, security and privacy of both the device and its wearer. Studies have shown that it is possible to authenticate a person through gait recognition and even through their heart beat. It would be interesting to study the ICMetric technology merged with the biometric security. Such a combination would create a system that can secure both the device and its owner.

Self-driving and smart vehicles get their intelligence from on-board computation and communication systems. A hurdle in the adoption of smart vehicles is their security. Smart vehicles are equipped with an overwhelming number of sensors and chips that monitor the vehicle behaviour at every instance. The ICMetric technology can use these sensors to offer security to these unique systems. A merger of ICMetric technology and smart vehicles has the ability to offer unprecedented security that cannot be promised with conventional cryptographic systems.

REFERENCES

[1] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *J. Supercomput.*, vol. 68, no. 1, pp. 1–48, Apr. 2014.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[3] J. Cecílio and P. Furtado, "Wireless Sensor Networks: Concepts and Components," Springer International Publishing, 2014, pp. 5–25.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[5] Cambridge University Press., *Cambridge essential English dictionary.* Cambridge University Press, 2011.

[6] A. Marrington, D. Kerr, and J. Gammack, *Managing Security Issues and the Hidden Dangers of Wearable Technologies*. IGI Global, 2016.

[7] N. Dhanjani, *Abusing the Internet of things: blackouts, freakouts, and stakeouts*. O'Reilly, 2015.

[8] Y. Xiao and Y. Pan, *Security in Distributed and Networking Systems*. WORLD SCIENTIFIC, 2007.

[9] D. Genkin, L. Pachmanov, I. Pipman, A. Shamir, and E. Tromer, "Physical key extraction attacks on PCs," *Commun. ACM*, vol. 59, no. 6, pp. 70–79, Jun. 2016.

[10] J. A. Halderman *et al.*, "Lest We Remember: Cold-Boot Attacks on Encryption Keys," *Communications of the ACM*, vol. 52, no. 5, ACM, p. 91, 01-May-2009.

[11] A. Stanoyevitch, *Introduction to Cryptography with Mathematical Foundations and Computer Implementations*. Chapman & Hall/CRC, 2010.

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

[13] J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.

[14] A. McEwen and H. Cassimally, *Designing the Internet of Things*. Wiley, 2013.

[15] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A Review," in *2012 International Conference on Computer Science and Electronics Engineering*, 2012, vol. 3, pp. 648–651.

[16] C. Yang, W. Shen, and X. Wang, "Applications of Internet of Things in manufacturing," in *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2016, pp. 670–675.

[17] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[18] A. K. Sood and R. J. Enbody, "Crimeware-as-a-service—A survey of commoditized crimeware in the underground market," *Int. J. Crit. Infrastruct. Prot.*, vol. 6, no. 1, pp. 28–38, 2013.

[19] L. Piwek, D. A. Ellis, S. Andrews, and A. Joinson, "The Rise of Consumer Health Wearables: Promises and Barriers," *PLoS Med.*, vol. 13, no. 2, pp. 1–9, 2016.

[20] S. H. Weingart, "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses," Springer Berlin Heidelberg, 2000, pp. 302–317.

[21] F. Koeune and F.-X. Standaert, *A Tutorial on Physical Security and Side-Channel Attacks*. Springer Berlin Heidelberg, 2005.

[22] V. Pasupathinathan, "Hardware-Based Identification and Authentication Systems," Macquarie University, 2009.

[23] A. G. Mason and M. J. Newcomb, *Cisco secure Internet security solutions*. Cisco Press, 2001.

[24] C. V. Anchugam *et al.*, "Classification of Network Attacks and Countermeasures of Different Attacks," in *Network Security Attacks and Countermeasures*, vol. 50, no. 1, IGI Global, 2016, pp. 115–156.

[25] E. Blumenthal and E. Weise, "Hacked home devices caused massive Internet outage," *USA Today*, 2016. [Online]. Available: http://www.usatoday.com/story/tech/2016/10/21/cyber-attack-takes-down-east-coast-netflix-spotify-twitter/92507806/. [Accessed: 13-Nov-2016].

[26] I. Zeifman, D. Bekerman, and B. Herzberg, "Breaking Down Mirai: An IoT DDoS Botnet Analysis," *Imperva incapsula Blog*, 2016. [Online]. Available: https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html.

[27] M. Kan, "Friday's DDoS attack came from 100,000 infected devices — PCWorld," *PCWorld*, 2016. [Online]. Available: http://www.pcworld.com/article/3135273/security/fridays-ddos-attack-came-from-100000-infected-devices.html.

[28] S. Yu, *Distributed denial of service attack and defense*. Springer-Verlag,

2014.

[29] D. K. Bhattacharyya and J. K. Kalita, *DDoS attacks : evolution, detection, prevention, reaction, and tolerance*. CRC Press, 2016.

[30] J. J. Stapleton, *Security without obscurity: A guide to confidentiality, authentication, and integrity*. CRC Press, 2014.

[31] I. Kizhatov, "Physical Security of Cryptographic Algorithm Implementations," Universite Du Luxembourg, 2011.

[32] A. Young and M. Yung, "The Dark Side of 'Black-Box' Cryptography or: Should We Trust Capstone?," Springer Berlin Heidelberg, 1996, pp. 89–103.

[33] A. Cullen and L. Armitage, "The social engineering attack spiral (SEAS)," in *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*, 2016, pp. 1–6.

[34] A. Rae and L. Wildman, "A Taxonomy of Attacks on Secure Devices," in *Proceedings of the Australia Information Warfare and Security Conference 2003*, 2003, pp. 251–264.

[35] M. Gupta, J. Walp, and R. Sharman, *Threats, Countermeasures and Advances in Applied Information Security*. IGI Publishing, 2012.

[36] L. J. Camp and M. E. Johnson, *The Economics of Financial and Medical Identity Theft*. Boston, MA: Springer US, 2012.

[37] N. Kshetri, *The Global Cybercrime Industry*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[38] I. X-force, "Reviewing a year of serious data breaches , major attacks and new vulnerabilities," 2016.

[39] V. Morabito, "Wearable Technologies," in *The Future of Digital Business Innovation*, Cham: Springer International Publishing, 2016, pp. 23–42.

[40] J. Hofdijk, B. Séroussi, C. Lovis, F. Sieverink, F. Ehrler, and A. Ugon, Eds., "Transforming Healthcare with the Internet of Things," in *Proceedings of the EFMI Special Topic Conference 2016*, 2016.

[41] J. Lindström, "Security challenges for wearable computing a case study," in *4th International Forum on Applied Wearable Computing (IFAWC),* 2007.

[42] S. C. Mukhopadhyay, Ed., *Wearable Electronics Sensors*, vol. 15. Cham: Springer International Publishing, 2015.

[43] P. Kumar and P. C. Pandey, "A wearable inertial sensing device for fall detection and motion tracking," in *2013 Annual IEEE India Conference, INDICON 2013*, 2013.

[44] A. Muro-de-la-Herran, B. García-Zapirain, and A. Méndez-Zorrilla, "Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.

[45] S. Sprager and M. B. Juric, "Inertial sensor-based gait recognition: A review," *Sensors*, vol. 15, no. 9, pp. 22089–22127, 2015.

[46] J. Chauhan, H. J. Asghar, M. A. Kâafar, and A. Mahanti, "Gesture-based Continuous Authentication for Wearable Devices: the Google Glass Case.," in *14th International conference on Applied Cryptography and Network Security*, 2016, pp. 1–28.

[47] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz, "A wearable system that knows who wears it," in *12th annual international conference on Mobile systems, applications, and services - MobiSys '14*, 2014, pp. 55–67.

[48] A. Burns *et al.*, "SHIMMER™ – A Wireless Sensor Platform for Noninvasive Biomedical Research," *IEEE Sens. J.*, vol. 10, no. 9, pp. 1527–1534, Sep. 2010.

[49] M. Rahman, B. Carbunar, and M. Banik, "Fit and Vulnerable: Attacks and Defenses for a Health Monitoring Device," 2013.

[50] J. R. Vacca, *Network and system security*. Syngress, 2014.

[51] M. Bhushan and M. B. Ketchen, "Variability," in *CMOS Test and Evaluation*, New York, NY: Springer New York, 2015, pp. 201–239.

[52] D. Merli and R. Plaga, "Physical unclonable functions: devices for cryptostorage," in *Proceedings of the 3rd international workshop on Trustworthy embedded devices - TrustED '13*, 2013, pp. 1–2.

[53] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security - CCS '02*, 2002, p. 148.

[54] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," in *Information Hiding*, Springer-Verlag, 2009, pp. 206–220.

[55] M. Deutschmann, "Cryptographic Applications with Physically Unclonable Functions," 2010.

[56] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[57] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual conference on Design automation - DAC '07*, 2007, pp. 9–14.

[58] P. Tuyls and B. Škorić, "Strong Authentication with Physical Unclonable Functions," in *Security, Privacy, and Trust in Modern Data Management*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 133–148.

[59] M. J. Atallah, E. D. Bryant, J. T. Korb, and J. R. Rice, "Binding software to specific native hardware in a VM environment," in *Proceedings of the 1st ACM workshop on Virtual machine security - VMSec '08*, 2008, p. 45.

[60] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis," *IEEE*

Trans. Comput. Des. Integr. Circuits Syst., vol. 34, no. 6, pp. 889–902, Jun. 2015.

[61] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, “Physical one-way functions.,” *Science (80-.).*, vol. 297, no. 5589, pp. 2026–30, Sep. 2002.

[62] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, “Design and implementation of PUF-based ‘unclonable’ RFID ICs for anti-counterfeiting and security applications,” in *2008 IEEE International Conference on RFID (Frequency Identification), IEEE RFID 2008*, 2008, pp. 58–64.

[63] G.-J. Schrijen and V. van der Leest, “Comparative analysis of SRAM memories used as PUF primitives,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2012, pp. 1319–1324.

[64] M. Cortez, A. Dargar, S. Hamdioui, and G.-J. Schrijen, “Modeling SRAM start-up behavior for Physical Unclonable Functions,” in *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2012, pp. 1–6.

[65] Intrinsic-ID Inc, “SRAM PUF: The secure silicon fingerprint,” 2016.

[66] K. B. Frikken, M. Blanton, and M. J. Atallah, “Robust Authentication Using Physically Unclonable Functions,” in *International Conference on Information Security*, 2009, pp. 262–277.

[67] H. Handschuh, G.-J. Schrijen, and P. Tuyls, “Hardware Intrinsic Security from Physically Unclonable Functions,” in *Towards Hardware-Intrinsic Security*, Springer Berlin Heidelberg, 2010, pp. 39–53.

[68] M. Deutschmann, “Cryptographic Applications with Physically Unclonable Functions,” Alpen-Adria-Universitat Klagenfurt, 2010.

[69] R. Maes, A. Van Herrewege, and I. Verbauwhede, “PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2012, pp. 302–319.

[70] F. Kohnhäuser, A. Schaller, and S. Katzenbeisser, “PUF-Based Software Protection for Low-End Embedded Devices,” in *International Conference on Trust and Trustworthy Computing*, 2015, pp. 3–21.

[71] V. J. Rathod, N. C. Iyer, and Meena S M, “A survey on fingerprint biometric recognition system,” in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 323–326.

[72] A. N. Kataria, D. M. Adhyaru, A. K. Sharma, and T. H. Zaveri, “A survey of automated biometric authentication techniques,” in *2013 Nirma University International Conference on Engineering (NUiCONE)*, 2013, pp. 1–6.

[73] A. Cooper *et al.*, “RFC 6973: Privacy Considerations for Internet Protocols,” 2013.

[74] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G.

Vigna, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting," in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 541–555.

[75] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert, "Rozzle: De-cloaking Internet Malware," in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 443–457.

[76] P. Eckersley, "How Unique Is Your Web Browser?," Springer Berlin Heidelberg, 2010, pp. 1–18.

[77] A. Kurtz, H. Gascon, T. Becker, K. Rieck, and F. Freiling, "Fingerprinting Mobile Devices Using Personalized Configurations," *Proc. Priv. Enhancing Technol.*, vol. 2016, no. 1, pp. 4–19, Jan. 2016.

[78] Threat matrix, "Device Fingerprinting and Fraud Protection Whitepaper," pp. 1–6, 2015.

[79] D. M. Kristol, "HTTP Cookies: Standards, privacy, and politics," *ACM Trans. Internet Technol.*, vol. 1, no. 2, pp. 151–198, 2001.

[80] M. Caselli, D. Hadžiosmanović, E. Zambon, and F. Kargl, "On the Feasibility of Device Fingerprinting in Industrial Control Systems," Springer International Publishing, 2013, pp. 155–166.

[81] G. Taleck, "Ambiguity Resolution via Passive OS Fingerprinting," Springer Berlin Heidelberg, 2003, pp. 192–206.

[82] R. Grimes, "An Introduction to Honeypots," in *Honeypots for Windows*, Apress, 2005, pp. 3–34.

[83] R. L. Krutz and R. D. Vines, *The CEH prep guide : the comprehensive guide to certified ethical hacking*. Wiley, 2007.

[84] C. Anton-Haro and M. Dohler, *Machine-to-machine (M2M) communications : architecture, performance and applications*. Woodhead Publishing, 2014.

[85] T. Kohno, A. Broido, and K. Claffy, "Remote physical device fingerprinting," in *2005 IEEE Symposium on Security and Privacy (S&P'05)*, 2005, pp. 211–225.

[86] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Publishing, 2010.

[87] F. Diez, D. Touceda, J. M. S. Camara, and S. Zeadally, "Toward self-authenticable wearable devices," *IEEE Wirel. Commun.*, vol. 22, no. 1, pp. 36–43, Feb. 2015.

[88] E. Papoutsis, "Investigation of the Potential of Generating Encryption Keys for ICMETRICS," University of Kent, 2009.

[89] S. Tahir and I. Rashid, "ICMetric-Based Secure Communication," in *Innovative Solutions for Access Control Management*, vol. 36, IGI Global, 2016, pp. 263–293.

[90] X. Zhai *et al.*, "Application of ICmetrics for Embedded System Security," in *2013 Fourth International Conference on Emerging Security Technologies*, 2013, pp. 89–92.

[91] Y. Kovalchuk *et al.*, "Investigation of Properties of ICmetrics Features," in *2012 Third International Conference on Emerging Security Technologies*, 2012, pp. 115–120.

[92] Y. Kovalchuk, K. McDonald-Maier, and G. Howells, "Overview of ICmetrics Technology – Security Infrastructure for Autonomous and Intelligent Healthcare System," *Int. J. u- e- Serv. Sci. Technol.*, vol. 4, no. 3, pp. 49–60, 2011.

[93] R. Tahir, H. Huosheng Hu, D. Dongbing Gu, K. McDonald-Maier, and G. Howells, "Resilience against brute force and rainbow table attacks using strong ICMetrics session key pairs," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2013, pp. 1–6.

[94] E. Papoutsis, G. Howells, A. Hopkins, and K. McDonald-Maier, "Integrating Feature Values for Key Generation in an ICmetric System," in *2009 NASA/ESA Conference on Adaptive Hardware and Systems*, 2009, pp. 82–88.

[95] R. Tahir, H. Tahir, and K. McDonald-Maier, "Securing health sensing using integrated circuit metric," *Sensors (Switzerland)*, vol. 15, no. 10, pp. 26621–26642, 2015.

[96] Maxim Integrated Products, "DS2411 Silicon Serial Number with VCC Input," Sunnyvale, 2011.

[97] S. Devadas, "Practical Applications of Physical Unclonable Functions (PUFs)." MIT - CSAIL IAP.

[98] N. Maluf and K. Williams, *Introduction to microelectromechanical systems engineering*. Artech House, 2004.

[99] V. Choudhary and K. Iniewski, *MEMS fundamental technology and applications*. CRC Press, 2013.

[100] K. Unger and J. Novak, *Game Development Essentials: Mobile Game Development*. Delmar/Cengage Learning, 2012.

[101] M. Kraft and N. White, *MEMS for automotive and aerospace applications*. Woodhead Publishing, 2013.

[102] S. Beeby, G. Ensell, M. Kraft, and N. White, *MEMS Mechanical Sensors*. Artech House, 2003.

[103] F. Renaut, "MEMS Inertial Sensors Technology," Swiss Federal Institute of Technology Zurich, 2013.

[104] O. Willers, C. Huth, J. Guajardo, and H. Seidel, "MEMS-based Gyroscopes as Physical Unclonable Functions," in *23rd ACM Conference on Computer and Communication Security*, 2016, no. 261, pp. 591–602.

[105] D. G. Senesky and B. Jamshidi, "MEMS strain sensors for intelligent structural systems," in *Lecture Notes in Electrical Engineering*, vol. 96,

Springer Berlin Heidelberg, 2011, pp. 63–74.

[106] C. P. Mayer, "Security and Privacy Challenges in the Internet of Things," *Electron. Commun. EASST*, vol. 17, 2009.

[107] R. Vemal, C. Lo, S. Ong, B. S. Lee, and C. C. Yong, "MEMS vs. IC manufacturing: Is integration between processes possible," in *2009 1st Asia Symposium on Quality Electronic Design*, 2009, pp. 39–43.

[108] C. Acar and A. Shkel, *MEMS vibratory gyroscopes: structural approaches to improve robustness*. 2008.

[109] D. J. Fonseca, M. Sequera, D. J. Fonseca, and M. Sequera, "On MEMS Reliability and Failure Mechanisms," *Int. J. Qual. Stat. Reliab.*, vol. 2011, pp. 1–7, 2011.

[110] H. Bojinov and Y. Michalevsky, "Mobile Device Identification via Sensor Fingerprinting," 2014.

[111] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable," in *Network and Distributed System Security Symposium (NDSS)*, 2014, no. February, pp. 23–26.

[112] A. Aysu, N. F. Ghalaty, Z. Franklin, M. P. Yali, and P. Schaumont, "Digital fingerprints for low-cost platforms using MEMS sensors," in *Proceedings of the Workshop on Embedded Systems Security - WESS '13*, 2013, pp. 1–6.

[113] G. Baldini, G. Steri, F. Dimc, R. Giuliani, and R. Kamnik, "Experimental Identification of Smartphones Using Fingerprints of Built-In Micro-Electro Mechanical Systems (MEMS)," *Sensors*, vol. 16, no. 6, p. 818, Jun. 2016.

[114] A. Karlsson, "Mobile Device Sensor Fingerprinting With A Biometric Approach," Linköpings universitet, 2015.

[115] ISO, *Statistics - Vocabulary and symbols. Part 2: Applied statistics. ISO 3534–2.* ISO, 2006, p. 125.

[116] T. Van Goethem, W. Scheepers, D. Preuveneers, and W. Joosen, "Accelerometer-Based Device Fingerprinting for Multi-factor Mobile Authentication," in *Engineering Secure Software and Systems*, 2016, pp. 106–121.

[117] P. Cappelletti and A. Modelli, "Flash Memory Reliability," in *Flash Memories*, Boston, MA: Springer US, 1999, pp. 399–441.

[118] Y. Wang, W. Yu, S. Wu, G. Malysa, G. E. Suh, and E. C. Kan, "Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints," in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 33–47.

[119] D. C. Leblanc, *Statistics: Concepts and Applications for Science*, vol. 2. Jones & Bartlett Publishers, 2004.

[120] M. de S. Joaquim P, *Applied Statistics Using SPSS, STATISTICA, MATLAB and R*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[121] A. G. Bluman, *Elementary statistics : a step by step approach*. McGraw-Hill, 2012.

[122] K. M. Ramachandran and C. P. Tsokos, *Mathematical statistics with applications*. Academic Press, 2009.

[123] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to probability and statistics*, 13th ed. Brooks/Cole, Cengage Learning, 2009.

[124] Freescale Semiconductor, “1.5g - 6g Three Axis Low-g Micromachined Accelerometer,” 2005.

[125] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, “A physically unclonable function with BER for robust chip authentication using oscillator collapse in 40nm CMOS,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, 2015, pp. 1–3.

[126] D. M. Tanner *et al.*, “The effect of humidity on the reliability of a surface micromachined microengine,” in *IEEE International Reliability Physics Symposium Proceedings. 37th Annual (Cat. No.99CH36296)*, 1999, pp. 189–197.

[127] T. G. Brown and B. S. Davis, “Dynamic high-g loading of MEMS sensors: ground and flight testing,” in *SPIE 3512*, 1998, p. 228.

[128] D. M. Tanner *et al.*, “MEMS reliability in shock environments,” in *IEEE International Reliability Physics Symposium*, 2000, pp. 129–138.

[129] S. L. Miller *et al.*, “Failure modes in surface micromachined microelectromechanical actuators,” in *1998 IEEE International Reliability Physics Symposium Proceedings 36th Annual (Cat No 98CH36173) RELPHY-98*, 1998, pp. 17–25.

[130] T.-V. Hoang, L. Wu, S. Paquay, J.-C. Golinval, M. Arnst, and L. Noels, “A study of dry stiction phenomenon in MEMS using a computational stochastic multi-scale methodology,” in *17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, 2016, pp. 1–4.

[131] P. Gauravaram, “Security Analysis of salt——password Hashes,” in *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, 2012, pp. 25–30.

[132] M. Bellare, R. Canetti, and H. Krawczyk, “Keying Hash Functions for Message Authentication,” Springer Berlin Heidelberg, 1996, pp. 1–15.

[133] H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication.” RFC Editor, 1997.

[134] M. S. Turan, E. Barker, W. Burr, and L. Chen, *Recommendation for Password-Based Key Derivation - Part 1: Storage Applications*, no. December. 2010, p. 14.

[135] R. K, “Advanced encryption standard (AES) encryption for Kerberos 5, Network Working Group, RFC 3962,” 2005.

[136] Apple Inc., “iOS Security,” 2015.

[137] F. Stajano and R. Anderson, “The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks,” Springer Berlin Heidelberg, 2000, pp. 172–182.

[138] Bloodshed Software, “Bloodshed Software - Dev-C++.” [Online]. Available: http://www.bloodshed.net/devcpp.html. [Accessed: 23-Dec-2016].

[139] J. Viega, M. Messier, and P. Chandra, *Network security with OpenSSL*. O’Reilly, 2002.

[140] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, “Rabbit: A New High-Performance Stream Cipher,” in *International Workshop on Fast Software Encryption*, 2003, pp. 307–329.

[141] J. Schaad and R. Housley, “Advanced Encryption Standard (AES) Key Wrap Algorithm,” 197, 2002.

[142] W. Diffie and M. Hellman, “New Directions in Cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[143] D. Dicarlo, “Random Number Generation: Types and Techniques,” Liberty University , 2012.

[144] S. Bandyopadhyay and R. Bhattacharya, *Discrete and continuous simulation : theory and practice*. 2014.

[145] R. McEvoy, J. Curran, P. Cotter, and C. Murphy, “Fortuna: Cryptographically Secure Pseudo-Random Number Generation In Software And Hardware,” *Irish Signals Syst. Conf.*, pp. 457–462, 2006.

[146] L. Dorrendorf, “Cryptanalysis of the Windows Random Number Generator,” The Hebrew University of Jerusalem, 2007.

[147] J. Kelsey, B. Schneier, and N. Ferguson, “Yarrow-160: Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator,” in *Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography (SAC ’99)*, 1758th ed., Springer Berlin Heidelberg, 1999, pp. 13–33.

[148] P. Ribenboim, *The Little Book of Bigger Primes*, 2nd ed. Springer, 2010.

[149] R. Solovay and V. Strassen, “A Fast Monte-Carlo Test for Primality,” *SIAM J. Comput.*, vol. 6, no. 1, pp. 84–85, Mar. 1977.

[150] M. O. Rabin, “Probabilistic algorithm for testing primality,” *J. Number Theory*, vol. 12, no. 1, pp. 128–138, Feb. 1980.

[151] OpenSSL Software Foundation, “BN˙generate˙prime.” [Online]. Available: https://www.openssl.org/docs/man1.0.1/crypto/BN˙is˙prime.html. [Accessed: 24-May-2016].

[152] J. Viega and M. Messier, *Secure programming cookbook for C and C++*. O’Reilly, 2003.

[153] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[154] S. Y. Yan, *Primality Testing and Integer Factorization in Public-Key Cryptography*, vol. 11. Boston, MA: Springer US, 2009.

[155] Euclid and J. Williamson, *The Elements of Euclid. With dissertations.* Clarendon Press: Oxford, 1781.

[156] F. Saidak, "A New Proof of Euclid's Theorem," *Am. Math. Mon.*, vol. 113, no. 10, pp. 937–938, 2006.

[157] J. P. Pinasco, "New Proofs of Euclid's and Euler's Theorems," *Am. Math. Mon.*, vol. 116, no. 2, pp. 172–174, Feb. 2009.

[158] P. R. Vijayalakshmi and K. B. Raja, "Performance analysis of RSA and ECC in identity-based authenticated new multiparty key agreement protocol," in *2012 International Conference on Computing, Communication and Applications*, 2012, pp. 1–5.

[159] D. Li, Y. Wang, and H. Chen, "The research on key generation in RSA public-key cryptosystem," in *Proceedings - 4th International Conference on Computational and Information Sciences, ICCIS 2012*, 2012, pp. 578–580.

[160] D. Bernhard, M. Fischlin, and B. Warinschi, "Adaptive proofs of knowledge in the random oracle model," *IET Inf. Secur.*, vol. 10, no. 6, pp. 319–331, Nov. 2016.

[161] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98*, 1998, pp. 209–218.

[162] H. R. Zhu, "Security Analysis of EV-DO System," in *Progress on Cryptography*, Boston: Kluwer Academic Publishers, pp. 181–186.

[163] M. Barbosa and P. Farshim, "Secure Cryptographic Workflow in the Standard Model," in *Proceedings of the 7th international conference on Cryptology in India*, 2006, pp. 379–393.

[164] A. Das and C. E. V. Madhavan, *Public-key cryptography theory and practice*. Pearson Education, 2009.

[165] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, Apr. 1984.

[166] M. Bellare, "Practice-Oriented Provable-Security," Springer Berlin Heidelberg, 1999, pp. 1–15.

[167] N. Bagheri, P. Gauravaram, M. Naderi, and S. S. Thomsen, "On the Collision and Preimage Resistance of Certain Two-Call Hash Functions," Springer Berlin Heidelberg, 2010, pp. 96–105.

[168] Y. Lao, B. Yuan, C. H. Kim, and K. K. Parhi, "Reliable PUF-Based Local Authentication With Self-Correction," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 2, pp. 201–213, Feb. 2017.

[169] K. Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security," *IEEE Wirel. Commun.*, vol. 11, no. 1, pp. 62–67, Feb. 2004.

[170] X. Zhai *et al.*, "Exploring ICMetrics to detect abnormal program behaviour

on embedded devices," *J. Syst. Archit.*, vol. 61, no. 10, pp. 567–575, Nov. 2015.

[171] E. Ben Sasson *et al.*, "Zerocash: Decentralized Anonymous Payments from Bitcoin," in *2014 IEEE Symposium on Security and Privacy*, 2014, pp. 459–474.