electronics

MDPI

Article

PUF-Based Key Generation Scheme for Secure Group Communication Using MEMS

Mubarak Mehdi [1,†], Muhammad Taha Ajani [1,†], Hasan Tahir [1,*,†], Shahzaib Tahir [2,†], Zahoor Alizai [1,†], Fawad Khan [2,†], Qaiser Riaz [1,†] and Mehdi Hussain [1,†]

[1] Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan; mmehdi.msis16seecs@seecs.edu.pk (M.M.); mtaha.msis16seecs@seecs.edu.pk (M.T.A.); zalizai.msis17seecs@seecs.edu.pk (Z.A.); qaiser.riaz@seecs.edu.pk (Q.R.); mehdi.hussain@seecs.edu.pk (M.H.)

[2] Department of Information Security, College of Signals, National University of Sciences and Technology, Rawalpindi 46000, Pakistan; shahzaib.tahir@mcs.edu.pk (S.T.); fawadkhan@mcs.edu.pk (F.K.)

* Correspondence: hasan.tahir@seecs.edu.pk

† These authors contributed equally to this work.

Abstract: Consumer electronics manufacturers have been incorporating support for 4G/5G communication technologies into many electronic devices. Thus, highly capable Internet of Things (IoT)-ready versions of electronic devices are being purchased which will eventually replace traditional consumer electronics. With the goal of creating a smart environment, the IoT devices enable data sharing, sensing, awareness, increased control. Enabled by high-speed networks, the IoT devices function in a group setting thus compounding the attack surface leading to security and privacy concerns. This research is a study on the possibility of incorporating PUF as a basis for group key generation. The challenge here lies in identifying device features that are unique, stable, reproducible and unpredictable by an adversary. Each device generates its own identity leading to collaborative cryptographic key generation in a group setting. The research uses a comprehensive hardware testbed to demonstrate the viability of PUFs for the generation of a symmetric key through collaboration. Detailed analysis of the proposed setup and the symmetric key generation scheme has shown that the system is scalable and offers unrivalled advantages compared to conventional cryptographic implementations.

Keywords: symmetric key; cryptography; Internet of Things; Physical Unclonable Function (PUF); Group Diffie–Hellman

Citation: Mehdi, M.; Ajani, M.T.; Tahir, H.; Tahir, S.; Alizai, Z.; Khan, F.; Riaz, Q.; Hussain, M. PUF-Based Key Generation Scheme for Secure Group Communication Using MEMS. *Electronics* **2021**, *10*, 1691. https://doi.org/10.3390/electronics10141691

Academic Editors: Ikram Rehman, Sara Paiva, Nagham Saeed, Waqar Asif, Alexey Vinel and Priyadarsi Nanda

Received: 22 May 2021
Accepted: 7 July 2021
Published: 15 July 2021

1. Introduction

Micro-Electro-Mechanical Systems (MEMS) are composed of microscopic parts and mechanical components that are designed to sense physical phenomena such as acceleration, rotation, strain, etc. The MEMS technology is rapidly being incorporated into portable devices, smart clothing, vehicles, chemical industries, healthcare systems etc. An area that has particularly caught the attention of system designers and embedded system engineers is the Internet of Things (IoT) [1]. Pervasive computing along with the availability of 3G/4G/5G networks has facilitated the adoption of devices which can be carried, implanted in the body, worn, and placed ubiquitously. Due to evolution of networks, the devices can transmit video and large quantities of data seamlessly. Cyber-physical systems have been realised in the form of smart devices and interconnections have been created in the IoT environment. The cyber-physical systems have had a profound impact across different verticals including education, banks, healthcare, etc [2,3] and have experienced significant improvement in terms of the services they are providing through the interconnection of devices. IoT devices are network enabled devices which can share data

and communicate with one another for increased insight. In general, IoT devices owe their success to embedded sensors and the availability of networks which support high transfer rates thus supporting communication/intelligence. With the rapid adoption of 4G/5G networks there will be reduced latency, better data rates (even in densely populated areas) which was not at all possible in 2G networks but had limited possibility via 3G networks.

The design of modern IoT systems has been well studied and innovations are regularly brought to light. A common characteristic of IoT devices is that they continuously monitor the physical environment and then communicate data through a network interface. Thus, IoT devices are group-oriented and their interaction has caused physical trust boundaries and the virtual trust boundaries to overlap. This in itself compounds the security of the IoT device and its user. An adversary is therefore able to attack the virtual world causing harm in the physical world and vice versa.

Conventional security algorithms use the secrecy of cryptographic keys as a root of trust. If a cryptographic key is captured, the entire crypto system can be compromised. The cryptographic keys are often stored on the system/device which can be seen as an easy target for an adversary. An attempt to increase the key size simply makes brute force difficult but does little to deter the attacker. To make the situation even more complex the adversaries can capture the cryptographic keys through a variety of attacks that do not even target the algorithm or the cryptosystem. Side-channel attacks target those areas of the system, whose security is often overlooked. A novel root of trust such as physically unclonable functions (PUF) [4] can address the problem of key theft as the keys are not stored on the device. Hence cryptographic keys are generated only when required and discarded thereafter. This serves as a deterrent since attackers are unable to target the key storage location.

This research studies the bias in a MEMS accelerometer as a PUF feature to form a device identifier that can be used for the generation of group cryptographic key. Here the concern lies in using device features that are unique, stable and reproducible by the device while being unpredictable for an adversary. The purpose of this research is to propose an IoT security scheme based on PUF that can be implemented in the group environment.

1.1. Contributions

This research makes the following contributions to the existing research carried out in this domain:

- This research studies the MEMS accelerometer as a suitable Physical Unclonable Function. To establish the suitability of MEMS PUF in the IoT security, a sensor testbed has been established that studies identical sensors statistically. Hence the MEMS accelerometers have been analyzed statistically to show that there are enough inter sample variances along with sufficient intra sample similarities.
- To provision security services via PUF, this study presents a novel symmetric key generation algorithm based on which groups of IoT devices can communicate. The group key generation scheme is based on using the inherent device PUF to create a device identity which leads to the creation of a group key. Thus, the participants in a group can communicate with each other using the PUF as a root of trust for the group. The key generation scheme has been studied for varying key sizes and group sizes.
- A contribution of the proposed system is that it eliminates the need for stored keys. By eliminating stored keys, issues related to key theft are greatly reduced therefore increasing the overall security and reliability of the established group communications. Therefore, an extensive security analysis is also performed to verify the security of this research.

The security scheme is applicable to IoT devices functioning in the group setting. Effort has been made to ensure that the proposed system has minimum footprint and is scalable for large groups.

1.2. Organization

The remaining paper is organized by first throwing light on the IoT ecosystem and the threat landscape. Popular application areas of IoT have been mentioned with examples. Section 3 throws light on cryptographic key theft and its possible eradication through a novel root of trust. The compound security setup present in group communications has been discussed in Section 4. The proposed PUF ID establishment and details of the test bed along with statistical analysis of MEMS sensors has been given in Section 5. The use of device identity for the establishment of symmetric group key has been provided in Section 6. The scalability of the proposed system and its technical analysis has been detailed in Section 7.

2. Internet of Things and the Threat Landscape

The Internet of Things (IoT) is a pool of physical devices that are interconnected via high-speed network connections as shown in Figure 1. The IoT is composed of smart devices equipped with a range of sensors that allow them to sense (physical, physiological, chemical occurrences) and communicate data autonomously between other devices and peers. This implies that the effective functioning of the IoT environment is heavily dependent upon the correct functioning of the embedded sensors. Diagram below depicts common IoT applications possible due to the many forms of network connectivity. The cloud supports analytics, insights for prediction, analysis, forecasting, usage monitoring etc.

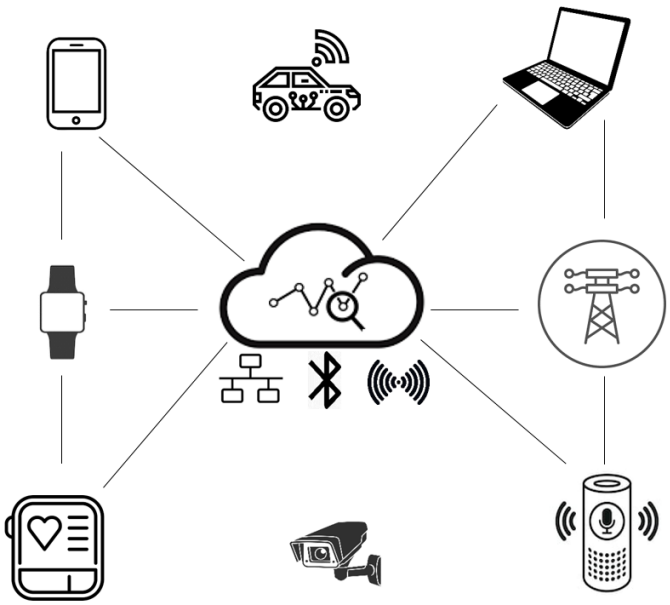

Figure 1. IoT applications.

Applications of IoT Enabled Smart Environments

Although the applications of IoT are limitless, here we limit our discussion to the most prominent and common applications. IoT is rapidly being used for communication and computation in an urban setting. Traffic management, utilities, disaster management, etc have been upgraded to make them smarter. The aim of smart cities [5] is to reduce expenditure, improve efficiency, increase safety and ultimately optimize public services. Similar uses have been seen in home automation applications, temperature control, appliance control, security system monitoring etc. The domain is greatly facilitated by virtual assistance devices [6] which can execute common tasks and process through voice command.

IoT devices are used for the detection of catastrophes such as earthquakes, storms, fires in homes and forests that could be life-threatening. Similar applications have been seen where scientists study shifts in weather patterns and other natural phenomena [7].

IoT devices are used for monitoring, measuring and management of utilities. A common use of these technologies is in smart grids to monitor electricity consumption [8]. The use of IoT in monitoring of utilities is used for billing and future demand prediction.

To increase crop yield and facilitate farmers, many IoT-based applications are being deployed [9]. IoT is being used to control climate conditions, monitor moisture in the soil and control temperature and environment. The monitoring and control of environmental metrics such as temperature and humidity can help farmers by monitoring the quality and health of crops preventing fungus and other microbiological infestations. The use of smart technologies in dairy farms has resulted in increased and sustainable dairy productions.

3. Cryptographic Key Theft

IoT is a computing paradigm that comprises of both humans and technology. The boundary between the physical and virtual worlds has overlapped thus amplifying security concerns that are present in the IoT. Research [3,10] has already brought to light numerous prevalent concerns related to privacy, data confidentiality, integrity, location tracking, user profiling, etc. Study [11] has shown that there are a variety of devices with varying purposes and capabilities in the IoT. The author has shown through experiments how IoT devices such as smart home lighting, baby monitors, electronic door locks, and smart TVs can be attacked to cause disruption of services and compromise cryptographic implementations. The author has shown that by analyzing the firmware of a prominent brand's smart-TV, the cryptographic key can be extracted. It has also been shown that all TV's of the same model have been programmed to work with a single key. This level of poor security implementation is particularly worrying for users.

Conventional cryptographic algorithms rely on the use of an intractable public algorithm for the provision of security services. Thus, the security of the entire system lies on the secrecy of the cryptographic key. According to the Kerckhoff's security principle, the security of a system lies in keeping the key secret and not the algorithm. Claude Shannon in [12] expressed a similar concept which states that the enemy knows the system. As compared to passwords that require an authenticatee (the entity to be authenticated) to provide authentication data; a solution to a complex mathematical problem is required in key-based authentication schemes for authentication [13]. In key-based authentication, no authentication information or keys (that are used as a substitute for passwords) are ever communicated. These schemes offer a higher security level by securing the authentication information against eavesdropping attacks.

Cryptographic keys can be large and complex owing to which they cannot be memorized. Thus, they are stored on a system for use in a cryptographic algorithm. There are numerous attacks that can capture keys thus leading system compromise [14]. Research has shown that keys can be captured through a range of side-channel attacks. An example of an invasive side-channel extraction is cold boot attack [15]. In this attack an adversary can cold boot to a lightweight operating system and then dump the RAM contents to removable storage. When a computer is powered off the RAM can retain data for a few seconds. To extend this duration to minutes and possibly hours, the RAM can be sprayed with cool air from a can of liquid nitrogen.

In another attack, researchers [16,17] have shown that it is possible to capture ElGamal and RSA cryptographic keys using an electromagnetic probing device that measures a narrow frequency band around the carrier. After subjecting the obtained signal to filtering, demodulation, distortion compensation and averaging, a clean aggregate trace is revealed which can be used to recover the key [11,18]. Such attacks have been made possible because the keys reside on a system and then loaded into memory or processed when required. The existence of the key on the system makes it susceptible to theft leading to system compromise. It is worth noting that these attacks and many more do not target the core cryptographic algorithm.

Incorporating a Novel Cryptographic Root of Trust

To counter threats related to key theft, this research has explored the use of MEMS sensors for the establishment of cryptographic keys. A novel root of trust that has recently gained much interest owing to the resilience it promises against common key theft attacks is physically unclonable functions (PUF). Fundamentally, a device PUF is a one-way function that is based on a system challenge-response. The challenge is chosen carefully so that it is easy to create and can provide a response that is unique, reproducible, robust and un-spoofable. These qualities are crucial for applications of PUF in cyber-physical systems. The PUF of a device is created using inherent device features. These features are introduced by fabrication, materials and environmental noise etc. One of the first studies [19] in PUF studied the placement of a static scattering medium in the path of a laser beam. It was discovered that the splatter pattern caused by the laser beam hitting the scattering medium is unique. The research was not adopted owing to low application potential. There are other types of PUF such as delay PUF, butterfly PUF, SRAM PUF, etc.

MEMS sensors are designed to be precise yet sensitive components. The accuracy of the MEMS sensors is impacted owing to many reasons, e.g., soldering a sensor onto the main board. When a sensor is soldered onto the main board, the resulting stresses influence the sensor functioning permanently. Here it is worth mentioning that the error introduced is not linear owing to which its eradication requires complex calibration algorithms. The calibration process is intended to rectify the inaccuracy in readings but does not eliminate the error. The residual bias in a MEMS sensor is a unique feature specific to a device. This study attempts to show that the bias of a MEMS sensor is a suitable PUF that has qualities including uniqueness, reproducibility, robustness and un-spoofability.

4. Problems in Sensor Group Communications

4.1. Dishonest Participants

Multiparty environment is composed of multiple devices communicating with each other. In multiparty environment, the presence of dishonest participants is one of the most important challenge in context to the security of the group key distribution. The distribution of the key in the presence of a dishonest participant can compromise the security of the group key. Many group key generation schemes that are available or widely used are weak and dishonest participants can take advantage by compromising the security of the group key. The scheme proposed in [20] is vulnerable to key theft attack. If a dishonest participant can connect itself to three different participants at the same time, the dishonest participant can derive the key. The scheme proposed in [21] requires precomputed certificates. If the dishonest participant can craft the packet with known plaintext or known cyphertext and forge the certificate, then the dishonest participant can create a key of its choice.

4.2. Dynamic Memberships and Forward/Backward Secrecy

In a dynamic group the number of participants can be changed, i.e., a member or members can leave or join the group. The membership of the group will not be the same so the same key cannot be used. If a member leaves the group, then he should not be able to decrypt the messages that were sent after he had left the group. This can be done by achieving forward secrecy, i.e., a new key should be generated whenever a member leaves the group. Similarly, if a new member is added to the group, that member should not be able to decrypt the old messages that were sent before that member joined the group. This can be done by achieving backward secrecy which means that a new key should be generated whenever a member is added to the group [22].

4.3. Single Point of Failure

A single point of failure (SPOF) is a component that upon failing would bring the entire system down. Fundamentally, this exists because of the system architecture layout which can cause potential failure due to a single failure point. For example, if an application requires users to login, then this can be a single point of failure. Kerberos is a network

authentication protocol known to have a single point of failure. It works using tickets that allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. If the central Kerberos server fails, no user can authenticate itself [23]. Hence schemes that are centralised by design are particularly prone to SPOF.

4.4. DDoS/DoS Attack

When the legitimate access to a resource or service is either denied or disrupted it is called Denial of Service (DoS). Distributed Denial of Service (DDoS) typically happens through multiple sources such as devices, computers and internet connections running in a coordinated fashion, to reduce service availability thus preventing legitimate user access to the service [24]. In DDoS the targeted server is flooded with bad requests or specially designed packets that will reduce the bandwidth of legitimate user. An attacker will send the especially designed packets to the application server that will cause the system to reboot or freeze. DDoS on group communication protocol can cause a halt to the flow of data. DDOS is a big threat especially in environments such as IoT network where the communication between IoT devices is end-to-end, that means a disruption might cause the delay of communication between multiple devices [25].

4.5. Collaborative Keying vs. Dictative Keying

Key generation is a delicate matter especially in multiparty or group environments. This is because there are no standard architectures mentioned for group key generation in the literature. Often group architectures are compared with social networking or chat application. The comparison of security-based group communication schemes with commercial applications is not correct because in security-based schemes the most important factor is whether the keying is collaborative or dictative. Below are two possible architectures for secure group communication key generation.

- **Dictative Key:** In dictative key approach, the responsibility of generating a key is given to a single or nominated participants. Normally Group Controllers (GC) or Key Generation Center (KGC) are responsible for generation of the key for the entire group. The problem with this approach is that the GC or KGC should be protected from the attacks because if the GC or KGC becomes compromised or are under D/DoS attack, then the security of group communication is also compromised. Thus, these type of schemes could lead to a single point of failure. Dictative key generation architecture is also not coherent with the philosophy of PUF because it will not take the input from the participants of the group.
- **Collaborative Key:** In collaborative keying approach, the participants of the groups are required to provide their inputs for generation of the key. In this key generation architecture, the risk of compromised GC or KGC is addressed because all the members are responsible for generating a group key and the GC/KGC are eliminated from the architecture. Another advantage of collaborative key generation architecture is that there will be no single point of failure and in case a member is not available its contribution will not be involved, and the key generation process will continue its works but would take the contributions from other participants of the group.

5. System Model

The research is applicable to devices that wish to communicate securely in a group setting. Suppose there are n devices that wish to communicate with each other securely. Due to an increase in attacks originating from compromised key distribution centers (and others mentioned in Section 4) an alternate is needed to establish a secure group. To achieve this the devices establish a secure group based on MEMS PUF. Here PUF allows the provision of cryptographic services that are based on inherent device features that are reproducible, unique and stable but unpredictable by an adversary. The devices create their own individual PUF "fingerprint" and provide contributions to establish a

collaborative symmetric group key. The group key is renegotiated whenever their is a change in membership and no individual device can force/dictate the group key.

5.1. PUF ID Establishment

The establishment of a PUF ID is a crucial phase of the entire scheme. In this phase the device features are identified that exhibit uniqueness across a large sample. Along with being unique, the features should also be repeatable. The flowchart showing individual phases is given in Figure 2.

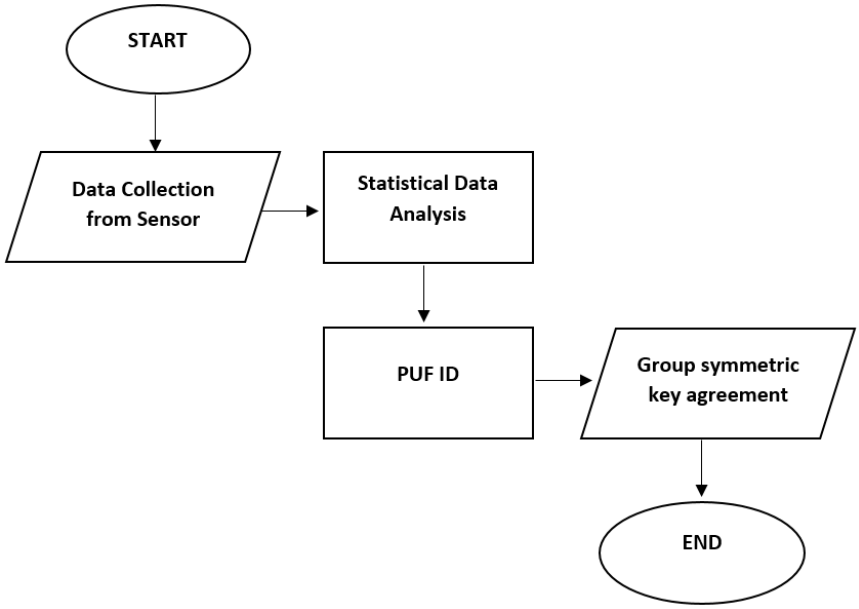

Figure 2. Flowchart showing individual phases.

5.2. System Implementation

This research is composed of two implementations, i.e., a PUF hardware test bed and a group symmetric key generation module.

The PUF hardware test bed is composed of MPU-6050 sensor [26]. The MPU-6050 is a MEMS sensor embedded with an accelerometer and gyroscope. The sensor has a 16-bit analog to digital converter which increases accuracy along the three-axis as values can be captured simultaneously. For precise tracking the sensor allows user-programmable full scale range of $\pm 2g, \pm 4g, \pm 8g$ and $\pm 16g$. To collect the MPU-6050 axis values an external Arduino UNO is used which makes it easier to program and give full control over sensor. The sensor testbed is composed of three identical hardware setups to test for the existence of identifying features. To test the existence of the PUF ID, the accelerometer sensors are subject to vibration-free and motion-free surface. To test for reproducibility, the sensor is subjected to this standard stimulus and the experiments are repeated under strict conditions.

A contribution of this paper is a group symmetric key agreement scheme and is discussed in Section 7. This module has been simulated and tested on a third generation Intel Core i5 3320M 2.60GHZ processor computer with 8GB RAM. The proposed scheme has been simulated in Java 1.8.0_121, while the platform used for development is NetBeans IDE 7.3.1.

5.3. Data Collection

To create a unique PUF ID, readings from each accelerometer are taken by providing a standard stimulus. To obtain the values of accelerometer the devices as shown in Figure 3 are placed in stable position and effort is made to ensure that there is minimum external influence such as vibration which can alter the sensor readings. Here it is important to

select a stimulus that can easily be created in the lab and by the user. Although the test bed is composed of three identical accelerometers, discussions and demonstrations have been limited to one accelerometer owing to limitation of space.

Readings of the three-axis x, y and z are captured to create the PUF ID. The sampling rate for the offset value varies from device to device; for our device sampling rate is 50 Hz. For every device, 10 samples were recorded for the individual axis. In a 10 s sampling window, 500 individual readings are collected which will be processed for creating a PUF ID. Due to limitation of space we limit the discussion to the generation and reproducibility of the PUF ID on only a single sensor.

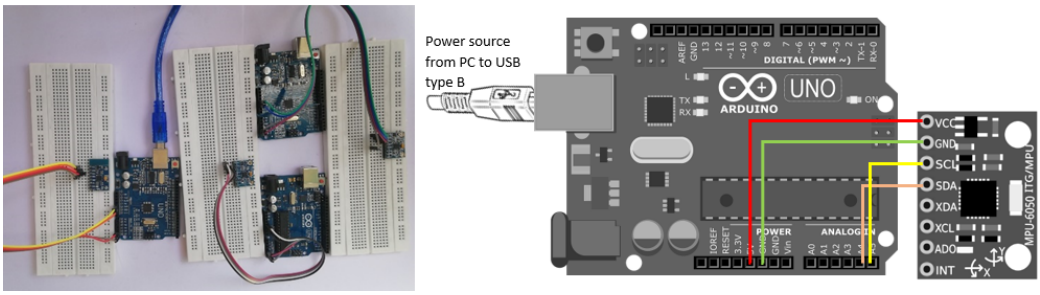

Figure 3. Hardware setup both physical and the schematic showing MPU-6050 connected to Arduino Uno.

5.4. Statistical Analysis of Collected Data

The recorded axes values are subjected to the Root Sum Square RSS as shown in Equation (1). This is a tolerance analysis method that assumes the normal distribution describes a variation of dimensions. The RSS calculated by adding the square of all three axes at a particular instance and then taking its square root. This step is repeated for all readings obtained from the sensors.

$$RSS_i = \sqrt{(x_i)^2 + (y_i)^2 + (z_i)^2} \tag{1}$$

Device fingerprints are practical if they are strictly unique and reproducible. The accelerometer sensor is subject to statistical tests to prove that readings obtained through multiple runs of a single device generated by a sensor are repeatable. Analysis of variance (ANOVA) is used for comparing the mean of two or more samples. ANOVA proves that the samples calculated are equal and there is no significant difference in the mean. ANOVA is distributed into three types:

1. A one-way analysis is used when three or more groups are compared based on a single factor
2. A two-way analysis is used when two or more groups are compared based on more than two factors.
3. A K-way analysis is used when the factor variables are K in number.

In ANOVA, p-value is an important factor; it is used to accept or reject a null hypothesis. The null hypothesis confirms that there is significant similarity in the data collected or the mean is same for collected data. The significance of p-value is that if it (the p-value) is less than 0.05 then null hypothesis is rejected. If p-value is greater than 0.05 then null hypothesis is accepted therefore leading to the conclusion that there is a significant similarity in the groups, or the mean is same for all groups.

As with all statistical parametric tests there are certain characteristics about the data which are known as assumptions. Violation of these assumptions changes the outcome of the parametric test. The following three ANOVA assumptions apply to the data collected from the experiment:

Assumption 1. *All the offsets values collected for every round are independent of each other and under the standard stimulus.*

Assumption 2. *The data collected must be normalized.*

To apply ANOVA, the data needs to be normalized as this is a prerequisite for the application of the test. To check the normality of the data, Shapiro–Wilk and Kolmogorov–Smirnov [27] test is performed on RSS using IBM SPSS tool [28]. According to Shapiro–Wilk test, if p-value is greater than α-value then data are normalized. As confidence interval is set to 95% so α-value is 0.05. The results in Table 1 show RSS 1 Kolmogorov–Smirnov p-value is 0.200 which is more than 0.05 and in Shapiro–Wilk test p-value is 0.204 which is also greater than 0.05; this confirms that when subjected to both tests RSS 1 is normalized. Similarly, both tests are applied on every other RSS. In RSS 7 and RSS 10 p-value is 0.033 and 0.029 which is less than 0.05 which states that sample is not normalized but in Shapiro–Wilk test values are 0.511 and 0.241 which is greater than 0.05. In a case if there is a contradiction in both the test results, Shapiro–Wilk test is preferred. Thus, according to the Shapiro–Wilk test all the RSS are normalized.

Table 1. Normality test results.

Device A	Normality Test					
	Kolmogorov–Smirnov			Shapiro–Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
RSS1	0.032	503	0.200	0.996	503	0.204
RSS2	0.029	503	0.200	0.997	503	0.567
RSS3	0.034	503	0.200	0.995	503	0.096
RSS4	0.022	503	0.200	0.997	503	0.503
RSS5	0.036	503	0.159	0.996	503	0.172
RSS6	0.036	503	0.155	0.966	503	0.298
RSS7	0.042	503	0.033	0.997	503	0.511
RSS8	0.032	503	0.200	0.997	503	0.548
RSS9	0.039	503	0.070	0.996	503	0.227
RSS10	0.043	503	0.029	0.996	503	0.241

Assumption 3. *In the obtained data, homogeneity of the variance has been obscured.*

IBM SPSS is used to perform the homogeneity of variance test. The result based on mean shows a p-value 0.088 higher than α-value 0.05 thus proving the hypothesis that all variances are homogeneous. Similarly, from adjusted degree of freedom df median and trimmed mean, it is also clear that p-value is higher, so assumptions are satisfied.

Table 2 shows the results based on mean, median, median with adjusted df and trimmed mean. Here $df1$ shows the total number of groups and $df2$ shows the total number of values from all the groups. Levene [29] statistics show the result generated applied on mean, median and sig. Here sig is the significance value or p-value that tells if there is homogeneity in the variance.

As all three assumptions are satisfied, now one-way ANOVA can be applied to the samples to compare the mean of population. By analyzing the ANOVA results, there is no significant difference in the mean as the p-value 0.270 is greater than the α-value 0.05. Furthermore, as $p > 0.05$, therefore null hypothesis is accepted, thus confirming that there is significant similarity in data collected.

Table 2. ANOVA homogeneity of variance using levene test.

Test of Homogeneity of Variances				
Device A	**Levene Statistic**	**df1**	**df2**	**Sig.**
Based on Mean	1.681	9	5030	0.088
Based on Median	1.676	9	5030	0.089
Based on Median and with adjusted df	1.676	9	4996.156	0.089
Based on Trimmed Mean	1.682	9	5030	0.088

It can be seen in Table 3 that Sig value or p-value is greater than 0.05 which means null hypothesis is accepted. This confirms that there is significant similarity in the data or the mean value of the group is same. In Table 3 the degree of freedom df between groups is 9 and within the groups is 5030. Sum of squares between groups is used to calculate the difference between the group mean. This is done by calculating the variation of each mean and the grand mean sum of squares within groups.

Table 3. Results for ANOVA.

ANOVA					
Device A	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Between Groups	0.000	9	0.000	1.233	0.270
Within Groups	0.157	5030	0.000		
Total	0.157	5039			

Plotting the RSS values generates a unimodal normal distribution. Further statistical analysis is performed by calculating mean, standard deviation, interquartile range, from the RSS values leading to the generation of the PUF ID as shown in Tables 4 and 5. The analysis of statistical measures proves the difference of these values for every device which makes a strong metric for establishment of PUF ID for a device. As shown in Table 4, for device A, the mean, standard deviation and interquartile range is calculated from RSS of first sample and gives 0.9827, 0.0054 and 0.0080. Similarly, for device B RSS values are calculated and mean is 1.2133, standard deviation is 0.0051 and interquartile range is 0.0072 as shown in Table 5. To create a PUF ID from statistical measures these values are added to make the final PUF ID.

Table 4. Statistical overview of mean, standard deviation and interquartile between groups for Device A.

Device A	**RSS1**	**RSS2**	**RSS3**	**RSS4**	**RSS5**	**RSS6**	**RSS7**	**RSS8**	**RSS9**	**RSS10**
Mean	0.9287	0.9286	0.9289	0.9288	0.9293	0.9293	0.9287	0.9287	0.9285	0.92880
Standard Deviation	0.0054	0.0055	0.0058	0.0057	0.0055	0.0052	0.0055	0.0054	0.0055	0.0052
Interquartile Range	0.0080	0.0076	0.0083	0.0079	0.0073	0.0071	0.007	0.0067	0.0072	0.0075

Table 5. Statistical overview of mean, standard deviation and interquartile between groups for Device B.

Device B	**RSS1**	**RSS2**	**RSS3**	**RSS4**	**RSS5**	**RSS6**	**RSS7**	**RSS8**	**RSS9**	**RSS10**
Mean	1.2133	1.2128	2.2135	1.2128	1.2134	1.2129	1.2131	1.2131	1.2132	1.2126
Standard Deviation	0.0051	0.0050	0.0051	0.0052	0.0052	0.0052	0.0050	0.0051	0.0050	0.0049
Interquartile Range	0.0072	0.0065	0.0068	0.0076	0.0072	0.0070	0.0070	0.0069	0.0069	0.00675

After adding these statistical values, a single value is obtained which will be used for PUF ID of the device. It can be seen that every sample has a small variation and is not exactly similar for every sample collected. Therefore, when the statistical values are added there will be a slight visible variation. To calculate the similarity in the form of percentage the following formula is used:

$$\frac{Approximate\,value - Exact\,value}{Exact\,value} * 100 \tag{2}$$

where the exact value is the first value calculated from sample 1 and approximate value is the value from repeat sample. From the formula it is clear that the value can be negative, so taking the absolute value to eradicate the negative sign, multiply positive value by 100, this will obtain the error in percentage when subtracted from 100. After repeated experiments accuracy was observed to be greater than 99% in every case. Figures 4 and 5 show the RSS values for device A and device B respectively.

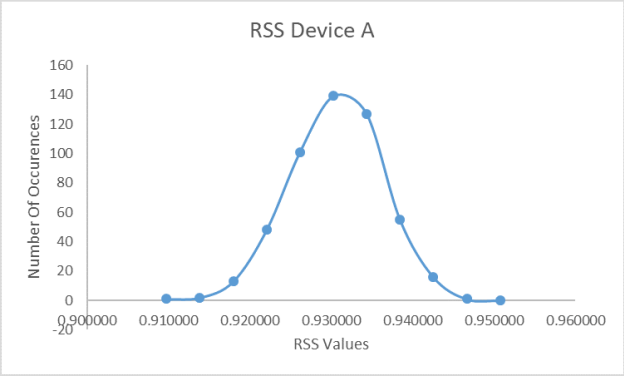

Figure 4. Root square Sum (RSS) for Device A.

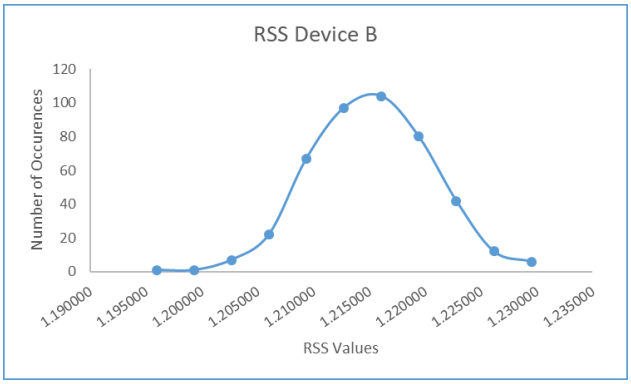

Figure 5. Root square sum (RSS) for Device B.

6. Group Key Agreement

Due to the increasing applications of group-oriented applications, the security of the collective group is considered an essential aspect of achieving confidentiality. The most important element in any cryptographic system is the key. If the key generation and key distribution process of any cryptographic system are weak, the resulting cryptographic system is considered to be vulnerable. Generating and distributing the keys is a difficult task especially in a group environment due to the possibility of having dishonest participants in the group. Additionally, distribution of the key over a large geographical area could pose risks.

A Group Key Agreement (GKA) protocol is a protocol where a group of members can agree upon a key in such a manner that the output of the algorithm is based on the contributions from all the members. This section throws light on the generation of symmetric keys in a GKA.

To establish secure group communications, constituent members of the group need to agree on cryptographic key(s). To accomplish this many schemes have appeared in research. The scheme proposed in [30] has a single point of failure, which means that if the central device is compromised or is not available then the scheme cannot continue functioning. It also requires large message sizes and precomputed certificates for the key agreement. The scheme proposed in [20] is vulnerable to key theft attacks. An attacker can derive the key if it manages to eavesdrop the message at three consecutive links in the conference network. An improved scheme proposed in [31] has overcome the issues which were seen [20] and has proven secure but is still not perfect for dynamic groups in which the members can join or leave the group.

The scheme proposed in this paper is inspired by the research presented in [31]. Our proposed work is a major improvement, as the original scheme proposed by the authors suffered from a significant design failure. To assist with the key generation as a result of dynamic membership, the last member of the group was responsible for ensuring key freshness whenever a membership change was seen. If this last member of the group leaves or his membership expires then key freshness cannot be achieved as the role of key generator was exclusively held by this last member of the group. Our proposed scheme is a considerable improvement as the communicating members hold only intermediate values which are used for the generation of the key. These intermediate values cease of exist once a session expires.

6.1. Proposed Scheme

The symmetric group key is generated by first taking the individual contributions from the group members. The contributions are based on unique individual identifications of each device/member. The three steps are as follows:

Stage 1. Create unique individual ID
Stage 2. A collection of individual contributions
Stage 3. Symmetric group key agreement

6.1.1. Create Unique Individual ID

In the first stage, all the members must create a unique secret R which will be used for calculating contributions. Each member can create his unique PUF identity PID based on his exclusive internal environment. Each member generates a random number using a random number generator $Rand()$. The random number is concatenated with the PID and the hash $h()$ is computed. The resulting value R is used in the upcoming stages. As every group member is in possession of his own unique value, therefore, it is denoted by R_i as follows:

$$R_i = h(PID||Rand()) \tag{3}$$

This equation forms the basis for the key generation scheme and any security provisions based on the PID. The $Rand()$ function is part of the cryptographic library and classed as a cryptographically secure pseudo-random number. The PID is a unique feature for every device thus ensuring that a hash of the two will result in unique hashes being generated every time. This serves as a key generation basis. As the proposed scheme targets dynamic group memberships therefore new keys are generated whenever there is a group membership change thus adding to the security of the scheme.

6.1.2. Contribution Collection

The second stage is to collect the contributions from all the members of the group by following Algorithm 1. In this stage, each member has to compute its share based on the values received from the previous member. Suppose there are P_i members actively communicating. If G is a large prime number used as an exponential base, assuming that group member P_4 receives a set of values $\{G^{R1R2R3}, G^{R1R2}, G^{R1R3}, G^{R2R3}\}$ from member P_3.

Member P_4 has to compute $\{G^{R1R2R3R4}, G^{R1R2R4}, G^{R1R3R4}, G^{R2R3R4}\}$ and send this to the next member P_5.

The pseudo-code given below is of a procedure used for collecting the contributions from the members of the group. The values required as input by this procedure are "G", "N", "R", and an array "$Previous$". G is a large prime number used as an exponential base, N is a large prime number used for order of the algebraic group (mod), R is the hash of PUF identity PID with a random number and "$Previous$" is an array of intermediate values received from the previous participant. In the case of the first participant, this array will be empty. The output will be a list of intermediate values "$Values$".

If the "$Previous$" is empty, then the "$Values[0]$" will be calculated using "$(G^R)\ mod\ N$" and the "$Values[1]$" will be equal to "$Values[0]$". If "$Previous$" is not empty, then the "$Values[0]$" will be equal to "$Cardinal$" which is "$(Temp^R)\ mod\ N$" and $Temp$ is equal to "$Previous[0]$" and "$Previous[1]$" will be equal to "$PreviousCV$". If the length of "$Previous$" $(Previous.Length)$ is equals to "2" then "$Value[2]$" calculated as "$(G^R)\ mod\ N$" else the remaining values of the list "$Values$" will be calculated as "$(Temp^R)\ mod\ N$".

Algorithm 1: TakeContribution.

```
Input: Bigint G, N, R, Previous[ ]
Output: Values[ ]
Values[Previous.Length + 1]
Cardinal←0, PreviousCV←0,
Intermediate←0, Temp←0
Temp ← Previous[0]
Cardinal←(Temp^R) mod N
Values[0]←Cardinal
PreviousCV←Temp
Values[1]←PreviousCV
if Previous.Length EQUALS 2 then
    Intermediate←(G^R) mod N
    Values[2]← Intermediate
else
    For i←2 TO Previous.Length, Temp←Previous[i-1]  Intermediate←(Temp^R)
     mod N
     Values[i]←Intermediate
end
RETURN Values
```

6.1.3. Symmetric Group Key Generation

The third stage is to compute the final symmetric key. In this stage, the final member of the group will broadcast all the intermediate values so that all the other group members can calculate the final key using their respective intermediate values following Algorithm 2.

Algorithm 2: CalculateFinalKey.

```
Input: BigInteger "IntermediateValueRelavent, R, N"
Output: BigInteger "FinalKey"
FinalKey ← (IntermediateValueRelavent ^R), mod, N
```

7. System Analysis

7.1. Key Size vs. Participant Size Analysis

As the symmetric key is generated for the group environment through collaborations therefore an analysis comparing both key size and participant size is important. The proposed key generation scheme has been tested with three key sizes, i.e., 160, 256, 512 bits. Table 6 presents a comparison of key size and participant size without any communication and queuing delays. Although the effect of participant size on key generation is obvi-

ous it must be mentioned that large sized groups close to 400 and 500 participants will not be very common except in larger enterprises, manufacturing facilities, hospitals, etc. For larger sized groups the impact of latency should be considered. Another caution at this stage is that a group setup that has frequent membership changes will cause the key agreement phase to be reinitiated frequently. This will imply that the system spends more time in key agreement phase and perhaps less time in actually using the key for secure communication. Moreover, a new key would action forward and backward secrecy for the group participants.

Table 6. Total time taken by the Group Key Diffie–Hellman scheme.

Key Size	Total Number of Participants	Total Time (Milliseconds)
160 Bits	100	2914.2
	200	5551.8
	300	8310.6
	400	15,753.8
	500	24,848.2
256 Bits	100	3559
	200	9953
	300	19,276.2
	400	33,401.6
	500	49,265.4
512 Bits	100	8546.6
	200	29,635.2
	300	80,917.6
	400	147,624.8
	500	236,857.2

7.2. Scalability Analysis

The proposed scheme for the creation of symmetric keys is composed of two components i.e., upflow function and key generation function. In the upflow function the individual group members supply their own secret inputs for the creation of the key in the key generation function. Simulation of the proposed scheme has shown that a majority of the processing time is consumed by the upflow function.

To test the scalability of the proposed scheme it has been analyzed by comparing the number of participants and execution time. Under ideal conditions the scheme should be able to accommodate increasing number of participants with minimum time requirements. The simulation is started with a group size of 100 participants and is increased to 500 with an increment of 100 participants. Analysis has shown that a group size of 100 participants requires 1499 milliseconds while larger groups of 500 participants require 22,797 milliseconds. The time required by the scheme for groups of up to 100 participants is reasonable and should not be a source of concern. In Figure 6, a graph showing a comparison of number of participants and execution time is given below.

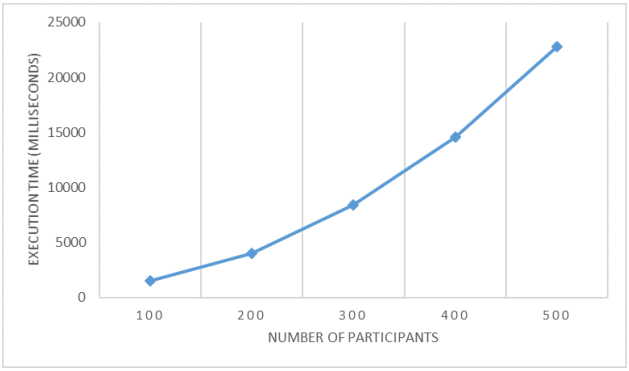

Figure 6. Graph showing a mean time comparison of number of participants and mean execution time.

The proposed scheme for the computation of the final key is composed of an upflow function that requires each participant to provide individual contributions. This function of the algorithm is particularly time-consuming as it is influenced by group size and runs collaboratively. In Figure 7 a comparison of the upflow function alongside the final key generation function is given.

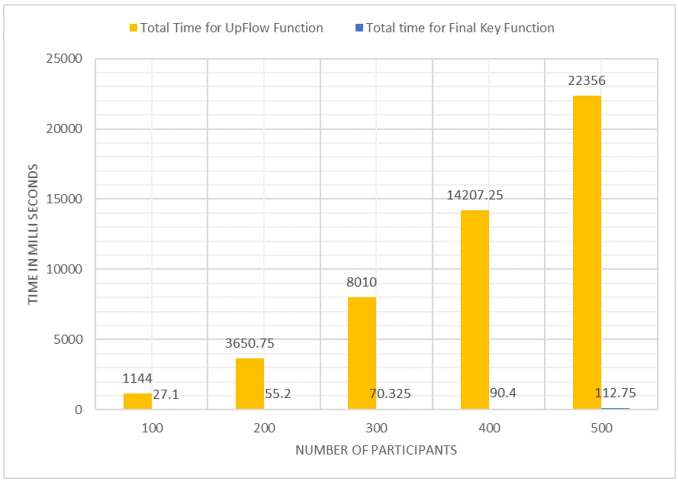

Figure 7. Graph showing a comparison of the upflow function alongside the final key generation function.

7.3. RAM Consumption Analysis

To show the effectiveness of the proposed schemes, they have been implemented and simulated for varying group sizes. Perhaps the greatest concern with security implementations targeting the group environment is the scalability of the proposed algorithms. When studying the RAM consumption it is worth mentioning that the schemes are not heavily influenced by the group size. This implies that increasing the group size does not have much impact on memory demand. Graph showing the relationship between group size and memory consumption is given in Figure 8.

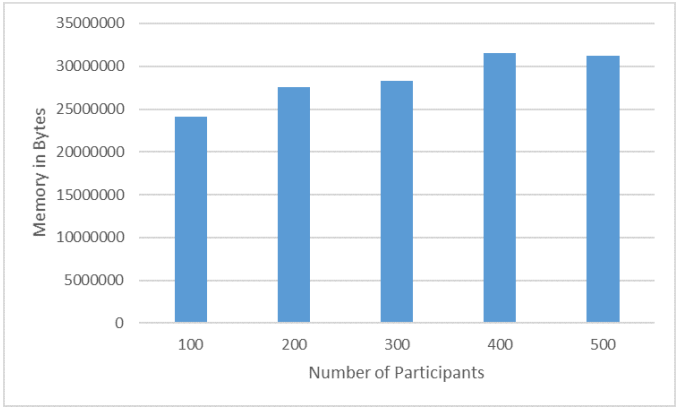

Figure 8. Graph showing the relationship between participant size and mean memory consumption for increasing group sizes.

7.4. Security Analysis

Communications in the group setting are more complicated therefore requiring increased security provisioning and implementations. As mentioned in Section 4, most concerns stem from dynamic memberships, dishonest participants, eavesdropping, MITM attack, key theft, compromised group controller etc. Elimination of these attacks and associated vulnerabilities at source is difficult as they exist mostly because of the group setting. In this research effort has been made to eliminate the above mentioned attacks which has largely been possible due to the advantages of the PUFs.

Key theft attacks are of many types, some of which are due to side-channel attacks. By incorporating PUF, a device is able to use its unique internal environment therefore creating a device identity. This identity is used to create a cryptographic key thus eliminating stored key theft attacks. Study of MEMS PUF has shown that sensor bias is a unique feature of the sensor and possesses properties of reproducibility, uniqueness, stability and unpredictability by an adversary. The bias is not a reproducible feature therefore making it difficult for an attacker to fabricate/spoof a device. Due to technological constraints it is not possible for even rogue manufacturers to clone a device with the same bias.

To eliminate the possibility of an attacker being able to guess the key through traffic analysis, a random salt is incorporated with the device identity and then hashed therefore forming the basis for the computation of a unique key each time it is needed. To protect the device identity, it is never shared in its true form.

The issue associated with dynamic memberships is **forward, backward secrecy and key freshness**. The proposed scheme facilitates dynamic membership and key freshness is assured since a fresh key is generated collaboratively every time there is a change in the group membership.

Establishing trust in the group environment can be difficult particularly in the presence of a third party. To prevent this, the scheme does not rely on a third party, instead the key is generated in a contributory manner and not dictated. This also eliminates **single point of failures** and issues with compromised GC/KGC.

Passive attacks can be very destructive as the adversary is capturing information through observation in a passive mode. The proposed key generation scheme is based on the intractable Diffie–Hellman Discrete Logarithm problem. As the key generation is collaborative and the individual contributions are never shared therefore it is not possible for a passive adversary to construct possessions of the group members. The scheme also prevents an adversary or a **dishonest participant** from forcing a key choice within the communicating parties.

Analysis has shown that active attacks are possible against the proposed scheme, but owing to the design the resulting impact on the system is limited. **Denial of service attack** is possible on the proposed scheme but successful conduct of MITM communication is not possible. A reason for this is that MITM cannot communicate bi-directionally in the multiparty setup. This can be better explained through the fact that each group member adds to the intermediate contribution values received from the uplink member. This is then passed onto the next participant. Hence a bidirectional communication flow is not provisioned in the proposed scheme. Thus, the impact of a MITM attack in key establishment is reduced to disruption of communications in the group.

7.5. Procedural Considerations

MEMS sensors are mechanical components known to possess characteristics that impact their adoption. Sensor aging and physical damage will have an impact on the readings obtained from the sensor. This is a known shortcoming of PUF-based sensors. Under normal use the impact is not sudden and can take possibly years to manifest (depending on applications). Similarly MEMS PUF is impacted by operational temperature, thus excessively high or low temperature could impact the resulting sensor readings. Both of the above should not be of concern when considering common IoT applications in an everyday environment.

Another concern to consider is that accelerometers can be fairly sensitive devices, i.e., they can pick up interfering vibrations. Thus, the presented scheme may suffer in an industrial setting where the sensors experience fatigue, excessive noise and vibrations. For the standard user they will need to place the sensor on a vibration-free surface under the standard stimulus to establish the root of trust for secure communications. If needed the issue of external vibrations can be corrected by incorporating vibration suppression components.

8. Conclusions

4G/5G ready versions of consumer electronics can now be purchased that leverage the power of the high-speed networks to create the IoT which is an environment based on sensing, increased connectivity and information sharing. Although users have already begun to reap the advantages offered by IoT systems, their wide adoption is often limited due to inherent security and privacy concerns. IoT devices are fundamentally group-oriented network devices which compounds the attack surface. A common threat faced by security implementations is that of cryptographic key theft. The cryptographic key is an important element that forms the basis of many security algorithms owing to which its secrecy and protection is imperative. The keys are often stored on a device which means that there are many possible attacks that could lead to their compromise. A novel root of trust that can offer resilience against key theft is physically unclonable functions (PUF). Fundamentally, these are one directional functions that are physical in nature and unclonable which makes them an attractive basis for security implementation.

This research has studied the use of MEMS PUF as a basis for group symmetric key generation. Thus, in this research a testbed has been established and the sensor bias is studied to show that it is a suitable feature in PUF-based key generation. Each device generates its own identity based on the internal PUF features. This identity is then used to compute a symmetric key for the group.

The novel symmetric key generation scheme is based on contributions from individual group members resulting in a symmetric key for secure group communications. To show the practicality of the proposed system it has been studied for scalability properties and a security analysis has also been performed. The symmetric keys have been tested for varying key and group sizes. The proposed system is an attempt at provisioning optimized security solution suitable for the IoT that resolves threats such as key theft, dynamic memberships, dishonest participants, side-channel key theft attacks.

The research has studied the provision of security via a PUF-based sensor testbed/proof of concept. Tests on a limited scale have suggested the feasibility of the study. Research on the topic is ongoing where the feasibility of the MEMS PUF will be tested on a larger number of sensors. The proposed schemes will also be tested on embedded smartphone accelerometers. This will facilitate scalable real life application testing.

Author Contributions: Conceptualization, H.T.; methodology, M.M. and M.T.A.; software, M.T.A.; validation, H.T., S.T. and Z.A.; formal analysis, S.T. and Z.A. and M.H.; investigation, F.K.; resources, M.M.; data curation, M.M. and Q.R.; writing—original draft preparation, M.M., M.T.A. and H.T.; writing—review and editing, S.T., Z.A. and Q.R.; visualization, F.K. and M.H.; supervision, H.T.; project administration, H.T.; funding acquisition, H.T. and Q.R. The authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially funded by National Center for Cyber Security (NCCS), Pakistan under the project titled "Privacy Preserving Search Over Sensitive Data Stored in the Cloud".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, R.; Wang, J. Internet of things: Application and prospect. In *MATEC Web of Conferences*; EDP Sciences, 2017; Volume 100, p. 02034. Available online: https://www.matec-conferences.org/articles/matecconf/pdf/2017/14/matecconf_gcmm2017_02034.pdf (accessed on 8 July 2021)
2. Catarinucci, L.; de Donno, D.; Mainetti, L.; Palano, L.; Stefanizzi, M.L.; Tarricone, L. An IoT-Aware Architecture for Smart Healthcare Systems. *IEEE Internet Things J.* **2015**, *2*, 515–526. [CrossRef]
3. Lee, J.; Bagheri, B.; Kao, H.A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manuf. Lett.* **2015**, *3*, 18–23. [CrossRef]
4. Alkatheiri, M.S.; Sangi, A.R.; Anamalamudi, S. Physical Unclonable Function (PUF)-Based Security in Internet of Things (IoT): Key Challenges and Solutions. In *Handbook of Computer Networks and Cyber Security*; Springer Publishing Company: Berlin/Heidelberg, Germany, 2020; pp. 461–473.
5. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 429–475. [CrossRef]

6. Kepuska, V.; Bohouta, G. Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 8–10 January 2018; pp. 99–103.
7. Ullo, S.L.; Sinha, G.R. Advances in Smart Environment Monitoring Systems Using IoT and Sensors. *Sensors* **2020**, *20*, 3113. [CrossRef] [PubMed]
8. Ghasempour, A. Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges. *Inventions* **2019**, *4*, 22. [CrossRef]
9. Friha, O.; Ferrag, M.A.; Shu, L.; Maglaras, L.; Wang, X. Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 718–752. [CrossRef]
10. Trappe, W.; Howard, R.; Moore, R.S. Low-energy security: Limits and opportunities in the internet of things. *IEEE Secur. Priv.* **2015**, *13*, 14–21. [CrossRef]
11. Dhanjani, N. *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts*; O'Reilly: Sevastopol, CA, USA, 2015.
12. Shannon, C.E. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
13. Alizai, Z.A.; Tahir, H.; Murtaza, M.H.; Tahir, S.; Mcdonald-Maier, K. Key-Based Cookie-Less Session Management Framework for Application Layer Security. *IEEE Access* **2019**, *7*, 128544–128554. [CrossRef]
14. Surendran, S.; Nassef, A.; Beheshti, B.D. A survey of cryptographic algorithms for IoT devices. In Proceedings of the 2018 IEEE Long Island Systems, Applications and Technology Conference, Farmingdale, NY, USA , 4 May 2018; pp. 1–8.
15. Yitbarek, S.F.; Aga, M.T.; Das, R.; Austin, T. Cold Boot Attacks are Still Hot: Security Analysis of Memory Scramblers in Modern Processors. In Proceedings of theInternational Symposium on High-Performance Computer Architecture, Austin, TX, USA , 4–8 February 2017; pp. 313–324.
16. Genkin, D.; Pachmanov, L.; Pipman, I.; Tromer, E. Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9293, pp. 207–228.
17. Genkin, D.; Pachmanov, L.; Pipman, I.; Shamir, A.; Tromer, E. Physical key extraction attacks on PCs. *Commun. ACM* **2016**, *59*, 70–79. [CrossRef]
18. Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the internet of things: A review. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 3, pp. 648–651.
19. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical one-way functions. *Science* **2002**, *297*, 2026–2030. [CrossRef] [PubMed]
20. Ingemarsson, I.; Tang, D.T.; Wong, C.K. A Conference Key Distribution System. *IEEE Trans. Inf. Theory* **1982**, *28*, 714–720. [CrossRef]
21. Harney, H.; Muckenhirn, C. Group Key Management Protocol (GKMP) Architecture. Available online: https://dl.acm.org/doi/pdf/10.17487/RFC2094 (accessed on 8 July 2021)
22. Rao, R.V.; Selvamani, K.; Elakkiya, R. A secure key transfer protocol for group communication. *arXiv* **2012**, arXiv:1212.2720.
23. Neuman, C.B.; Ts'o, T. Kerberos: An Authentication Service for Computer Networks. *IEEE Commun. Mag.* **1994**, *32*, 33–38. [CrossRef]
24. Bhatia, S.; Behal, S.; Ahmed, I. Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions. In *Advances in Information Security*; Springer: New York, NY, USA, 2018; pp. 55–97.
25. Zhang, C.; Green, R. Communication security in internet of thing: Preventive measure and avoid DDoS attack over IoT network. In Proceedings of the 18th Symposium on Communications & Networking, Alexandria, VA, USA, 12–15 April 2015; pp. 8–15.
26. InvenSense Inc. MPU-6000 and MPU-6050 Product Specification Revision 3.4 MPU-6000/MPU-6050 Product Specification. 2013. Available online: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf (accessed on 8 July 2021)
27. Hanusz, Z.; Tarasińska, J. Normalization of the Kolmogorov–Smirnov and Shapiro–Wilk tests of normality. *Biometrical Lett.* **2015**, *52*, 85–93. [CrossRef]
28. Leech, N.L.; Barrett, K.C.; Morgan, G.A. *IBM SPSS for Intermediate Statistics: Use and Interpretation*, 5th ed.; Routledge: London, UK, 2014.
29. Joaquim, P.; Marques, S. *Applied Statistics Using SPSS, STATISTICA, MATLAB and R*, 2nd ed.; Springer Publishing Company: Berlin/Heidelberg, Germany, 2007.
30. Goldwasser, S. *Advances in Cryptology—CRYPTO' 88: Proceedings*; Springer Publishing Company: Berlin/Heidelberg, Germany, 1990.
31. Steiner, M.; Tsudik, G.; Waidner, M. Diffie-Hellman key distribution extended to group communication. In Proceedings of the ACM Conference on Computer and Communications Security, New Delhi, India, 14–16 March 1996; pp. 31–37.