Covenant Journal of Informatics & Communication Technology. Vol. 9 No. 1, June, 2021
ISSN: *print* 2354 – 3566 *electronics* 2354 – 3507 DOI:

An Open Access Journal Available Online

ETEASH-An Enhanced Tiny Encryption Algorithm for Secured Smart Home

Olushina Raphael Oluwade[1], Olayemi Mikail Olaniyi[2], Yunusa Simpa Abdulsalam[3], Lukman Adewale Ajao[2], Francis Bukie Osang[4]

[1]Department of Computer Science, Federal University of Technology, Minna, Nigeria
[2]Department of Computer Engineering, Federal University of Technology, Minna, Nigeria
[3]Department of Computer Science and Engineering, University Mohammed VI Polytechnic, Ben Guerir, Morocco
[4]Computer Science Department, Faculty of Sciences, National Open University of Nigeria.
oluwade.pg717576@st.futminna.edu.ng, mikail.olaniyi@futminna.edu.ng, abdulsalam.yunusa@um6p.ma ajao.wale@futminna.edu.ng, fosang@noun.edu.ng

Received: 06.01.2021 Accepted: 14.05.2021
Date of Publication: June, 2021

Abstract— The proliferation of the "Internet of Things" (IoT) and its applications have affected every aspect of human endeavors from smart manufacturing, agriculture, healthcare, and transportation to homes. The smart home is vulnerable to malicious attacks due to memory constraint which inhibits the usage of traditional antimalware and antivirus software. This makes the application of traditional cryptography for its security impossible. This work aimed at securing Smart home devices, by developing an enhanced Tiny Encryption Algorithm (TEA). The enhancement on TEA was to get rid of its vulnerabilities of related-key attacks and weakness of predictable keys to be usable in securing smart devices through entropy shifting, stretching, and mixing technique. The Enhanced Tiny Encryption Algorithm for Smart Home devices (ETEASH) technique was benchmarked with the original TEA using the Runs test and avalanche effect. ETEASH successfully passed the Runs test with the significance level of 0.05 for the null hypothesis, and the ETEASH avalanche effect of 58.44% was achieved against 52.50% for TEA. These results showed that ETEASH is more secured in securing smart home devices than the standard TEA.

Keywords/Index Terms—Internet of Things, Cryptography, Smart home, Tiny encryption algorithm, and Pseudo random number generators

1. Introduction

In recent years, the Internet of Things (IoT) with its potentials has positively affected every aspect of human endeavors, ranging from manufacturing, agriculture, healthcare, businesses, logistics, government, cities, and homes (Raggett, 2016). IoT generally refers to scenarios where network connectivity and computing capability extend to objects, sensors, and everyday items, which might not necessarily be considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention (Rose et al., 2015; Buenrostro et al., 2018).

Smart home as an area of application of IoT, inclusive of the smart city, community, and grid (Alagbe et al., 2019), is a network of connected devices in human living environment, which communicate remotely with the inhabitants to raise their living and quality of life, the efficiency of energy consumed and their safety. The smart city is a network of connected cities, locations, or infrastructures to improve the quality of life (Souza et al., 2020). The wireless broadcast nature of the smart home makes it vulnerable to series of attacks such as eavesdropping, replay attacks, and man in the middle (Zhang et al., 2017). This era of connected devices, the world is drifting to, makes IoT a worthwhile subject to build a career on (Misra, 2020).

Examining existing works, for securing the smart home, a microcontroller driven by GSM module and handset, with access password, alerting homeowners of any intrusion with the aid of SMS, buzzer and LED was used (Hasan et al., 2015). The access control could be brute-forced. Zandamela (2017) proposed a system that makes available the security reports of the home to the homeowner, through GSM mobile technology and Arduino. This proposed solution is a reactive one. Abu et al (2018) equipped home devices with Passive Infrared and Infrared sensors which capture intruders' motion and sends it to the homeowners. This proposal does not have access control.

Most work examined proposed solution that is reactive instead of preventive, and that could be brute-forced. This work proposes a system that works, not minding the presence of adversaries. IoT is ubiquitous partly as a result of its usage of the existing technologies such as Wireless Sensor Networks, Radio Frequency Identification (RFID) (Oluwu et al., 2020), and Cloud computing which serves as a ready platform for its communication. Traditional cryptographic security measures cannot be applied to the low-capacity devices known as constrained IoT devices. Flexible security infrastructure is hence needed for the IoT such as Lightweight Cryptographic algorithm (Usman et al., 2017). Lightweight cryptographic algorithms are designed to function in constrained devices such as sensors, health-care devices, RFID tags, and contactless smart cards, securing them through cryptographic keys. The strength of a cryptographic key depends on the degree of how hard it is to guess. The degree of hardness of a key depends on the degree of randomness used in generating the key (Vassilev & Hall, 2014). Generating random numbers is essential to cryptography, and generating true random information is the most crucial cryptographic algorithm to effectively aid in security key management (Hughes & Nordholt, 2016).

This work through generated cryptographic keys addresses issues of vulnerabilities and weakness of TEA to enhance its security capability against threats and security issues in smart devices in the smart home environment. This paper is structured in five sections. Section 1 introduces the work, section two examines related works, section 3 gives the methodology and techniques used in this work, while section 4 discusses the result and section 5 concludes and gives recommendations.

2. Related Works

The smart home is one of the applications of IoT. There are several smart devices designed and developed based on IoT. IoT provides smart home devices the ability to sense, monitor, and record pertinent data about home environments. Characteristically, the smart home can be conceptualized into three namely (Kumar & Mittal, 2019): 1) Comfort, 2) Monitoring, and 3) Security, as shown in Figure 1.

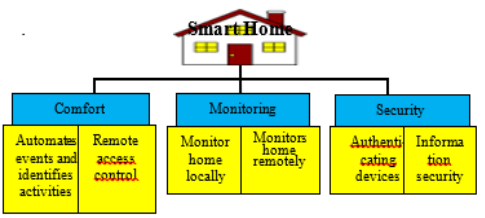

Figure 1. Smart home categorization into fields using services

In the "monitoring category," the homeowner can monitor the home and the environment remotely and able to call for assistance if there is an intrusion. The last category has to do with securing the home, the devices, and access to confidential information through authentication, access control, and encryption processes. It is estimated that 80% of smart devices are vulnerable to spectrums of attacks ranging from privacy, data, and identity theft, Permanent Denial of Service (PDoS), man-in-the-middle, Distributed Denial of Service (DDoS) to device hijacking (Rambus, 2019). All these spectrums of security challenges are categorized into three broad categories, namely: Confidentiality, Integrity, and Availability (CIA), known as CIA of information security (Mohammed et al., 2017). Figure 2 shows the CIA architecture for information security.

2.1 Data Grouping

The smart home as an application of IoT is faced with series of threats and security challenges jeered at compromising homeowners' privacy (Yassein et al., 2019). A three-layer IoT architecture also called the basic layer was proposed by Zeadally et al., (2019). These layers are: application, network, and sensing layers, and these layers are responsible for processing, transmission, and perception respectively. Figure 3 illustrates the Basic layer or three-layer IoT architecture.

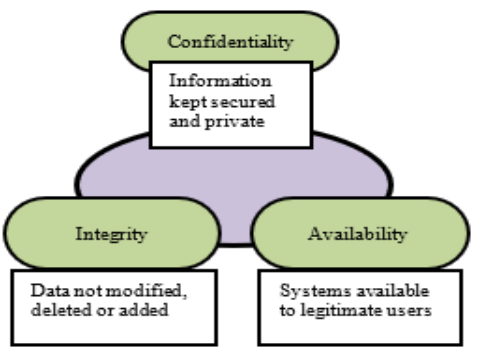

Figure 2. Architecture of CIA

In IoT and smart home environments, information and data are communicated by users and smart devices through unsecured public communication channels which are linked with the internet (Ajao et al. 2018). These exposed the smart home to several cyber-attacks and threats such as Denial of

Service (DoS), Distributed Denial of Service (DDoS), Permanent Denial of Service (PDoS), man-in-the-middle, privacy, data, and identity theft (Rambus, 2019). Other possible threats to smart homes are replay attacks, device cloning, device tampering, privacy breach, information disclosure, signal injection, spoofing, routing, and signal injection (Ali et al., 2019). Security in IoT and smart home is examined using Figure 4, based on three-layer IoT architecture ranging from sensing or perception layer (IoT devices), then, network layer (communication channels) and application layer (cloud computing, intelligent traffic, and smart home) (Zeadally et al., 2019). The three layers and their peculiar securities challenges are thus discussed in the next sub-section.

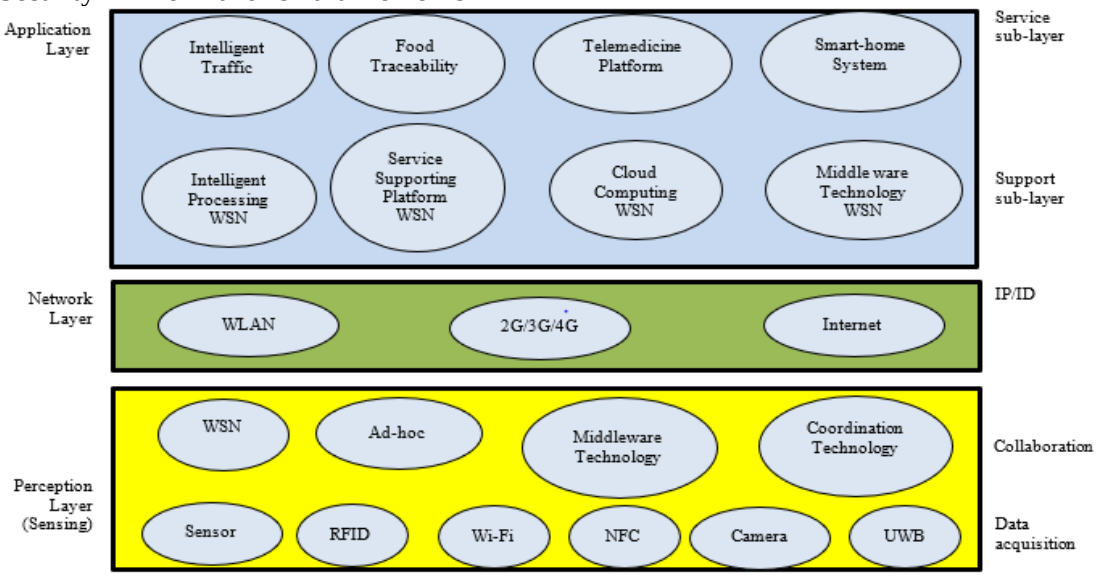

Figure 3. Basic Layer of IoT Architecture

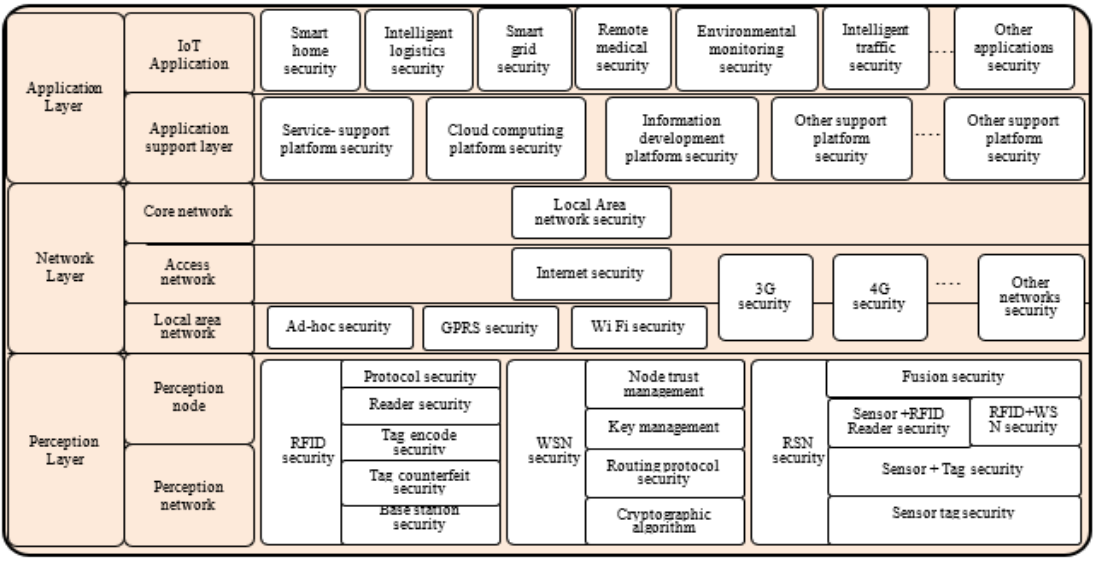

Figure 4. IoT Security Architecture

2.2 Data Gathering

In this layer, sensors collect information about, or of users. The sensors are attached to the materials or objects to collect information as sound, vibration, location, humidity, and motion as the need arises (Burhan et al., 2018). Some of the sensors or technology used here, to collect information are; wearable devices, RFID, WSN, Global Positioning System (GPS), Robust Secure Network (RSN), video games, and surveillance camera (Rao et al., 2018), (Burhan et al., 2018), (Popoola et al., 2017). The information collected by the sensors at this layer can be shared or hacked by hackers for illegal purposes. Some of the threats and attacks at this layer (Burhan et al., 2018) are:

1. Node Capturing: This is one of the deadly attacks that can be meted on the perception layer. It's a scenario when a node is captured, and the hacker can have access to confidential resources and information on the node. If control of a key node such as a gateway is taken over by a hacker, then information leakage between nodes may occur. Key used for secure communication and information on authentication and access codes stored in the gateway's memory may also be compromised.
2. Fake and Malicious Node: This is an attack where the attacker adds a fake node in the Home Area Network (HAN) with fake inputs to stop the transmission of real data. The fake node consumes the vital energy of normal nodes on the network and ultimately halts or destroys the network.
3. Eavesdropping: This is a situation where an attacker in real-time intercepts unauthorized communication such as fax communication, text messages, video conferencing, and calls as a result of unsecured transmission over the network.

4. Timing Attack: This occurs in devices whose computing capabilities are not strong; hence attackers can explore their vulnerabilities by studying how long it takes for the devices to respond to queries received.
5. Replay Attack: This attack occurs when a hacker eavesdrops on the communication between sender and receiver. Authentic information and identity can be seized from the sender to deceive the receiver. The correctness of information sent by the hacker may not be discovered by the receiver, since the information is encrypted.
6. Node cloning: This is a scenario where the identification information of a node or RFID tag is captured. The captured node or tag identification information is then compromised and can be replicated posing serious threats to the smart home network (Suchitra & Vandana, 2016).

2.3 IoT Network Layer and Its Security Challenges

This layer is also known as the transmission or transport layer, serving as a connector between the perception layer and the application layer (Burhan et al., 2018). It transmits the information sensed by sensors which are physical objects through communication channels that could be wired or wireless. This layer is responsible for linking smart devices, network devices, and networks to each other. With this make-up, the network layer is vulnerable to series of threats and attacks relating to authentication and integrity of information. Some of the common security threats here are:

1. Man-in-The-Middle Attack: This is when an attacker intercepts and interferes with the communication between two communicating smart devices getting confidential information (Rao et al., 2018).
2. Denial of Service (DoS) Attack: This attack hinders the legitimate users of smart devices and network resources, the access to services and network resources by flooding the target devices or resources with redundant or legitimate-like requests to make the device or resources inaccessible by authenticated users (Burhan et al., 2018) (Shaikh et al., 2019).
3. Wormhole Attack: In this attack, a tunnel is formed between two malicious nodes which enhance the transmission of packets between the two nodes. These two nodes feigned closeness which deceives neighboring nodes to forward packets through them. This leads to more time-consuming because of the longer time of packet transmission (Bhosale and Sonavane, 2018). It may also lead to packets or information hijacking.
4. Sybil Attack: The attacker creates several multiple fake identities to manipulate real nodes on the network. attack.
5. Spoofing: In a spoofing attack, the genuine node is blocked by a hacking node, pretending to be the genuine node. The hacking node waits for an opportunity to capture the credentials for the genuine node's authentication.

The created pseudo identities subvert the identification of nodes and thereby facilitating access of malicious nodes to the network (Alharbi et al., 2018). There is a threat to the privacy of connected nodes and users on the network in the Sybil

The genuine device or user's credentials stealthily captured can now be used to control the genuine device (Ling et al., 2017).

6. Gateway Attack: This attack is aimed at establishing disconnection between the nodes and the internet facility. This attack could also be routing or DoS in nature, where false or empty information is received by the node from the internet (Rao et al., 2018).
7. Spear phishing (Gupta, et. al., 2018): with this phishing attack, attackers pose as authentic company owner by using some of the features of the authentic and target sites to trick customers into giving out their personal and confidential information.

2.4 Application Layer and Its Security Challenges

The application layer hosts all applications in which IoT is deployed. Some of the applications in which IoT is deployed are smart home, smart health, smart city, and smart grid. The Application layer provides varying services to the various IoT applications, depending on the type of information gathered by the perception or sensing nodes of the Perception layer (Burhan et al., 2018). Smart home as an application of IoT has threats and vulnerabilities associated with it. Hence the Application layer has security challenges that are from within and without. The main security challenge threats could affect life, data, and information confidentiality, and integrity. Implementation of security in the smart home environment is a key issue because of its constrained nature. Some of the security challenges in this layer are:

1. Malicious Code Attack: The attacker injects malicious code into a smart device to have total control of the IoT system (Abdul-Ghani et al., 2018).
2. Phishing Attacks: This attack is aimed at getting access to users secured sensitive data, through access to their passwords, credit card, and other confidential information. This could be achieved through email hacking by sending fraudulent mails emails that appear to have originated from a trusted known source (Abdul-Ghani et al., 2018).
3. Software Defenselessness: This is when non-standard software written by a programmer is used within the network. This could increase the probability of vulnerabilities of the entire network and high chances of hacker's attack (Rao et al., 2018).
4. Social Engineering: In social engineering, the users in IoT setup are manipulated into disclosing their sensitive and confidential information for attackers' malicious acts (Abdul-Ghani et al., 2018).

This paper addresses smart home security in the context of information security. Several works in literature addressed the issue of smart home security. In Hossain et al., (2014) three requirements were provided to access smart devices in a Smart home: NFC tag, Password, and PIR motion sensor. Entrance into the home without the fulfillment of the three implies intrusion, and an alert is made by a high-intensity buzzer. The anomaly of this proposed work is that it does not have remote notification or remote alerting. This implies that, while not at home, and there is an intrusion, the homeowner would not be alerted.

Hasan et al., (2015) proposed securing a home with a microcontroller driven by a GSM module and handset, access password, alerting homeowners of intrusion with the aid of SMS, buzzer, and LED. The access control, a 4x4 keypad, and a mobile phone could be brute forced or compromised respectively.

Also, Zandamela, (2017) proposed a system that makes available the home security reports to the homeowner in absence with the aid of GSM mobile technology, Arduino, and the internet. The combination of these three made a monitoring website to be set up to receive real-time videos to monitor intrusion, motion, and fire outbreak. It could be observed, that the proposed is a reactive measure, reacting to incidence occurrence instead of preventive.

Abu et al., (2018) in their work equipped home devices with Passive Infrared (PIR) and Infrared (IR) sensors which capture motions and send them to the server through Blynk, an application on the devices. The sensed data received is sent to an internet-enabled microcontroller, which alerts the homeowner in real-time. The observation on this work is the absence of provision for access control. Pavithra (2019) proposed an automated monitoring system, that monitors the home for humans' presence and sends an alert message and call to Global System for Mobile Communication (GSM) phone number. This intrusion is also displayed on a Liquid Crystal Display (LCD). This work focused mainly on monitoring the home for intrusion. Another observation in this work is the use of a GSM phone number for alerting the homeowner. If the homeowner lost this phone number, then he would be unable to monitor the home. This could be disastrous to sudden time-critical events. In their work Karimi & Krit, (2019) proposed two methods that can be used for smart home security to address threats other than methods used by those examined in this literature review that made use of microcontrollers, GSM, and smartphones. The proposed methods are 1) Data encryption and 2) Network monitoring. Karim & Krit (2019) on "data encryption" suggested a combination of algorithms or developing a new algorithm for better security of data in transmission, while on network "monitoring," they proposed implementation of Intrusion Detection System (IDS), usage of smart gateway and Internet Service Provider (ISP) having firewalls.

Having reviewed works on smart home security that had to utilize, microprocessors, GSM modules, phones, sending SMS or calls, at securing the smart home, contributions have also been made in the context of user authentication and data encryption in literature. The smart home is a conglomeration of heterogeneous devices such as sensors, medical devices, and RFID tags which are constrained devices, having little or no memory capacity, low-capacity energy source or battery-powered and low processing capability (Jordi et al., 2017), communicating with one another, unaided by humans. These constrained devices or smart devices would require flexible security infrastructure such as Lightweight cryptography to cater to their low resources (Oluwade et al., 2018).

Cryptography is a technique traditionally used in securing data on transit or at rest stored in storage devices. Traditional cryptography cannot secure low-capacity IoT devices, called constrained devices; hence lightweight cryptography was designed to be used on constrained devices (Oluwade, et al., 2018). Key generation,

management, and encryptions are important aspects of cryptography ((Zhang et al., 2017). Shaikh et al., (2019) in their work pointed out that solution to eavesdropping and unauthorized insertion of messages to a network are message authentication, public-key cryptography, and symmetric key encryption techniques.

This work proposed an enhancement to the traditional Tiny Encryption Algorithm in a manner to secure home by securing information not minding the adversaries through information encryption compared to existing works which require four digits as a passcode to access the smart home, or use of SMS to alert the homeowner of intrusion to the home. The use of a passcode poses security risks to the home because the username or passcode could be predicted through brute-force attack and SMS might delay in delivery to alert of intrusion or might not be delivered at all due to bad network of the telecommunication company.

An existing work that lags when compared with this work made use of home monitoring videos sent over the internet to the homeowner wherever he is (Das et al., 2011). Kim et al., (2011) used social networking platforms to gain access to the home. Authors in El-Shafee & Hamed, (2012) use a username with the password to monitor and control home appliances. Home monitoring could be looped over the internet if cameras are not secured, while hackers could spoof relationships on social networking platforms to gain access to the home. Username with the password could be predicted and hence the security of the home could be compromised. The summary of related works is in Table 1. The confidentiality of information in smart homes is investigated in this work by hardening encryption keys through entropy generation to provide enough security for the smart home. The strength of a cryptographic key depends on the degree of how hard it is to guess. The degree of hardness of a key depends on the degree of randomness used in generating the key (Vassilev & Hall, 2014). Generating random numbers is essential to cryptography, and generating true random information is the most crucial cryptographic algorithm to effectively aid in security key management. This work improves on Tiny Encryption Algorithm (TEA) by overcoming its notable drawback of weak crypto keys.

2.5 Tiny Encryption Algorithm (TEA)

To efficiently ensure security in the smart home, a lightweight cryptographic algorithm is needed because it is specifically designed for constrained devices. There are lightweight cryptographic algorithms that may be used on constrained devices to produce good security but with less processing power, time, and memory than symmetric cryptographic algorithms such as PRESENT, HEIGHT, TEA, Advanced Encryption Standard (AES), and RC5 (Shifa et al., 2019). TEA stands out of the listed symmetric algorithm because of its short software codes aside from its other features which other symmetric algorithms have. This feature of shortcodes made TEA suitable for the development of embedded systems (Abdulsalam et al., 2018). However, with these strong features of TEA, it has a weakness, that should be corrected for its strength to be fully optimized.

The weakness of TEA is in its simplicity of encryption keys, which makes it vulnerable to equivalent key attacks (De Leon et al., 2019). This work aimed at enhancing TEA

to improve its security performance. Tiny Encryption Algorithm (TEA), originally designed by Wheeler and Needham is a small (tiny), and fast simple cryptographically strong algorithm with short software codes and small footprint as regards its memory occupation when stored in a device and seamlessly fit into any program on any computer (Shoeb & Gupta, 2013). These features of TEA made it suitable for security enhancement on constrained devices, which are appliances or nodes in IoT and smart home networks (De Leon et al., 2019).

The TEA's characteristics of block cipher, simplicity in codes of few lines, and implementation were achieved by making simple and weak, its basic operations (Shoeb & Gupta, 2013). Security challenges are overcome by repeating the basic operations repeatedly (Shoeb & Gupta, 2013). A credit to TEA is its high speed in encryption processes, but notable drawbacks of TEA are its use of "equivalent keys" which weakens its key length effectiveness and only requires complexity O(2ˆ32) which is even much lesser than the effort 2ˆ128 required for brute force attack to break the key.

The other notable drawback is that there is no known standard to which TEA is measured as regards the codes' length (Shoeb & Gupta, 2013).

The TEA is a Feistel cipher whose operation utilizes mixed algebraic groups, which are XOR, ADD and SHIFT. This operation utilizes the twin properties of Shannon – diffusion, and confusion which are important for the block cipher. TEA encrypts 64bits data at a time by using a 128-bit key and it's highly resistant to differential attacks. Related key attacks are possible with TEA, though its mixing portion seems to be okay (Kumar et al., 2015).

In TEA's basic operation, 32 rounds of TEA would be completed for every encrypted 64-bit block. Diffusion works to hide any statistical information between the plaintext and the ciphertext that may serve as a backdoor to the attackers, while confusion ensures that the statistical information between the ciphertext and the encryption key is kept secure, to thwart any effort of the attackers to discover the key (Abdelhalim et al., 2013). Diffusion and confusion are achieved in Feistel Cipher Structure as illustrated in Figure 1 through the use of substitution and permutation.

The substitution (addition, XOR'ing, and shifting) operation is performed on the left (L) of the plaintext, while Permutation operation is swapping at every round, of both halves of the plaintext. TEA has a 128-bit key which is divided into four, to give 32-bits keyword length K[0], K[1], K[2], K[3] as in Figure 5) that work on a 64-bit data block that is split into two 32-bit blocks called L and R (Left and Right side of the data block). The operations in the first half of the first round of TEA as in Figure 5 are: R has a left shift of 4 and is then added to K[0], it is added to Delta [(√5-1)*231], and passes through a right shift 5 and is added to K[1].

The three XOR operation applied to their result to obtained R for the next Feistel round because at this time, swap is carried out on R and L. Figure 5 shows a Single Round TEA with 2 Feistel Operations.

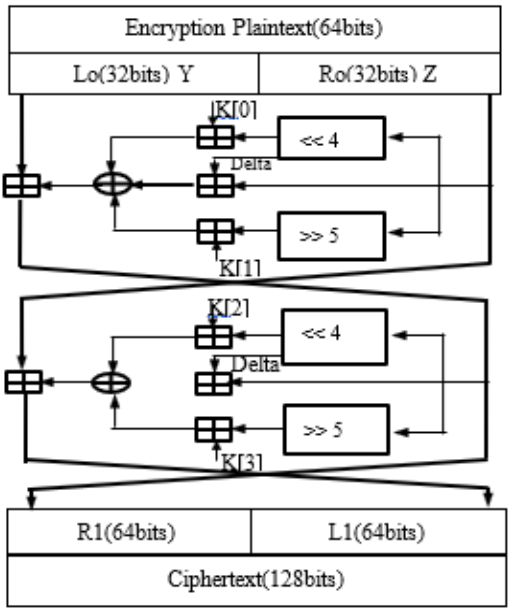

Figure 5. A Single Round TEA with 2 Feistel Operations

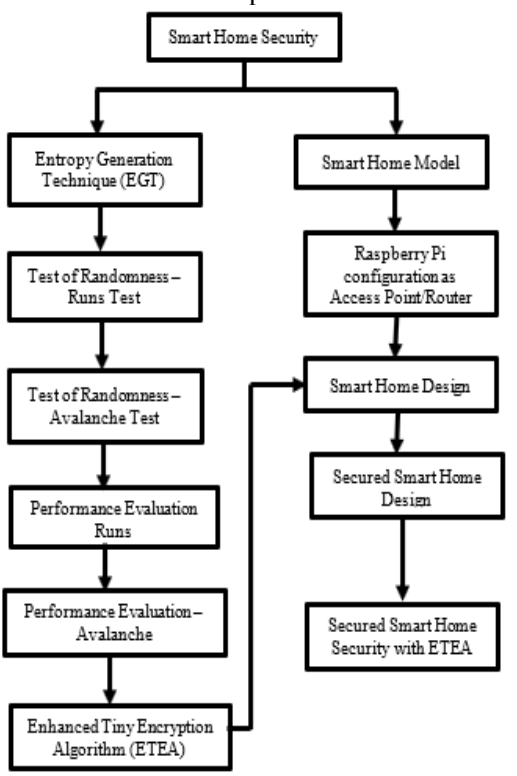

Figure 6. Taxonomy of the Research Methodology

Table 1. Summary of Related Works

S/No.	Author(s)/Work	Strength	Weakness	Remark
1.	i. Das, *et al.* (2011). ii. Kim, *et al*. (2011).	a. Utilized GPRS, Client-server communication with iOS application on phone. Motion sensors, cameras used to notify intrusion through internet b. Utilized social network platforms, SMS logs and call logs, for security	a. Videos of home could be looped by hackers over the internet if cameras are not secured properly b. Hackers could spoof relationship on social networking platforms to gain access to the home	a. Device hijacking attack. b. Unauthorized access Attack
2.	Elshafee and Hamed (2012).	WIFI communication, PC server,	Username and password	Prone to Brute force attack
3.	Hasan *et al*., (2015).	Securing home with access password and alerting homeowners of intrusion.	Key lock combination could be brute-forced	Brute-force attack
4.	Zandamela, (2017).	Sends real-time video and GSM-based information of home intrusions, motion and fire detections	This is just corrective measures.	Unauthorized access Attack
5.	Abu *et al*., (2018).	Monitors intrusion by sending alert report to the home owner	No provision for access control	Node tapering and Physical damage attacks
6.	Pavithra, (2019)	Automatically monitors human presence and sends alert through Mobile communication GSM	Loss or weak communication network and loss of mobile phone renders the system inoperative.	Physical Damage attack
7.	Karimi & Krit, (2019)	Data encryption	This is good	Data encryption is good

3. Methodology

The methodology involved the development of Entropy Generation Technique (EGT), ETEASH, and configuration of raspberry pi as router or access point and integration of modem/dongle on the raspberry pi. The methodology is segmented into Entropy Generation Technique, the Modula steps taken in achieving the technique, the algorithms involved in the technique, and the integration of the EGT to TEA giving Enhanced Tiny Encryption Algorithm (ETEA). The smart home was designed using raspberry pi3 as the access point for the home network. The next segment addresses the application of ETEA to the smart home, giving secured smart home, with a generic name: "Enhanced Tiny Encryption Algorithm for Secured Smart Home" (ETEASH) while the last segment is the tests carried out on EGT and ETEASH. The taxonomy of the methodology in realizing smart home security is in two divides as depicted in Figure 6. The divides are TEA enhancement and raspberry pi enhancement or configuration. TEA enhancement divide starts with Entropy Generation Technique, to, the test of the randomness of random numbers generated. EGT is incorporated into TEA to give ETEA. Performance evaluation was carried out on ETEA, compared with TEA. On the raspberry pi enhancement divide, the smart home was designed with the aid of a configured raspberry pi 3 as an access point or router. Incorporation of ETEA from the TEA enhancement divide to the designed smart home on the raspberry pi enhancement divide yielded a secured smart home, resulting in an Enhanced Tiny Encryption Algorithm for secured smart home (ETEASH). Figure 7 shows Smart Home secured with the Enhanced TEA (ETEA), with the ETEA components spelled out as EGT and TEA.

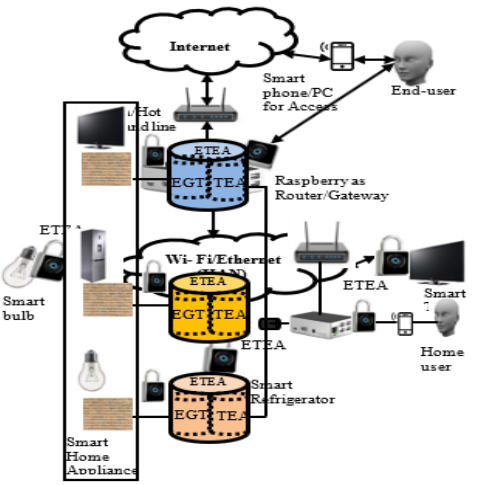

Figure 7. Smart Home Secured with Enhanced TEA (ETEA)

3.1 A Secured Smart Home and Configuration

The smart home with smart appliances connected with Wi-Fi, in a network known as Home Area Network (HAN) is shown in Figure 8. Raspberry Pi3 serves as the access point/router for the HAN. Modem/hotspot/Ethernet connects the raspberry to the internet; this invariably connects the entire smart home to the internet. The Raspberry Pi 3 was used in this research as a router or access point to

which all the smart devices are connected wirelessly.

The software that made this possible: HostAPD and ISC-DHCP-server are installed on the raspberry pi 3. A smartphone is used for controlling and monitoring the smart home. The secured smart home architectural design is illustrated in Figure 8. This is made up of the smart home and ETEA which is the smart home security component. The smart device messages or credentials are encrypted, and so with the integration of ETEASH, the smart home is secured from attacks. The smart hub, which is the raspberry pi 3 is the point of reference for data flow in the secured smart home. The directions of arrows from Figure 9 show the flow of data from one device to the other. The bi-directional arrows imply that the device can communicate, and it can be communicated to.

3.2 Hardware Components Design

The hardware components used consist of the raspberry pi 3, modem/dongle, and smartphone. The interconnection of these hardware devices gave the research work a physical structure. The raspberry pi 3 is the major hardware in this research, and the configuration is as shown in Table 2, while Table 3 shows the modem/dongle specification.

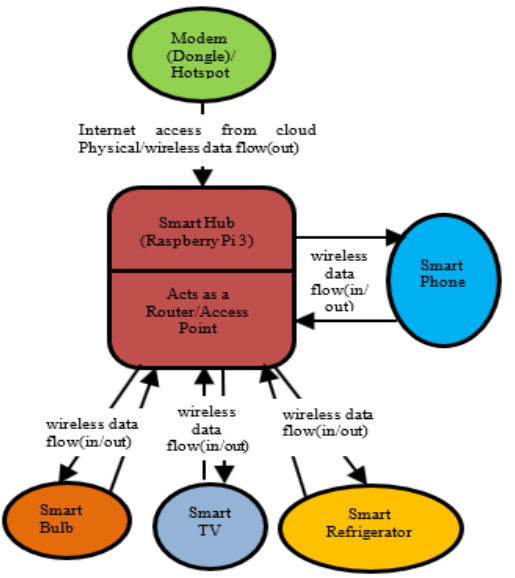

Figure 9. Data Flow between Smart Components

The modem/dongle serves as the link between the internet and the home area network. It provides internet service to the raspberry. Hotspot from a mobile device with internet service could be used alternatively. The smartphone or a personal computer (PC) as hardware serves as an access control unit from which communication to the smart devices are carried out.

Table 2 Raspberry Pi 3 Specification

S/NO	FEATURES	FEATURES'
1	Soc	BCM2837
2	CPU	Quad Cortex A53 @
3	Instruction	ARMv8-A
4	RAM	1GB SDRAM
5	Storage	Micro-SD
6	Ethernet	10/100
7	Wireless	802.11n/Bluetooth
8	Video Output	HDMI/Composite
9	Audio Output	HDMI/Headphone
10	GPIO	40

Table 3 Modem/Dongle Specification

S/NO	FEATURES' DETAILS
1	HSUPA USB Stick
2	Model MF110
3	ZTE Corporation

3.3 Entropy Generation Technique

The Java programming language codes and C++ codes were the internal components of the Entropy Generation Technique (EGT). The random number generating tools were carefully selected from the two programming languages. Pseudorandom number generating tools in C++ and C, are: rand() and srand(). Anytime encryption is made, four random numbers of 32 bits (concatenated to give 128 bits) are randomly picked from the generated random numbers, to serve as the encryption keys in the ETEASH.

Mathematically, for an attacker to be able to break the hardness of the encryption keys, if 2000 random numbers were generated, will be:

i. $2^{\wedge 2000}$ for the pick of 1st encryption key from 2000 randomly generated keys
ii. $2^{\wedge 2000}$ for the pick of 2nd encryption key from 2000 randomly generated keys
iii. $2^{\wedge 2000}$ for the pick of 3rd encryption key from 2000 randomly generated keys"
iv. $2^{\wedge 2000}$ for the pick of 4th encryption key from 2000 randomly generated keys"

Hence, the total probability $\sum_{i=1}^{4} P\left(K_i\right)$ of breaking the encryption keys of ETEASH for all four keys is: $2^{\wedge 2000}$ x $2^{\wedge 2000}$ x $2^{\wedge 2000}$ x $2^{\wedge 2000}$.

The functionality of rand() in generating entropies yields a similar sequence of entropies, while srand() yields a better random sequence of numbers, which leads to the adoption of srand() for Pool1 (C++). There are four random number generating tools in Java, which are: java.util.random, java.security.SecureRandom, /dev/urandom and OpenSSL*API. SecureRandom was adopted for the Pool2 (Java) component because it is cryptographically strong. Three steps of procedures are involved in EGT to have ETEASH developed, to enhance its hardness. These are: "Entropy shifting, stretching, and mixing"

1. *Module 1 - Entropy Shifting*

The theory behind entropy shifting is the movement of entropy from a pool 1 (storage 1) of entropy to another pool 2 (storage 2) of

entropy. This movement causes depletion of entropy in pool 1 and increment of entropy in pool 2. Java written codes serve as pool 1, while codes in C++ serve as pool 2. Entropy is generated using the random number generation function in C++ seeded with the system's time function. The generated entropy in pool 1 is then Entropy stretching" and "Entropy mixing". The steps are discussed in Module 1, Module 2, and Module 3 respectively. The design flowchart for the EGT is shown in Figure 10. Two rounds of random numbers generation from pool 1 and pool 2 are ORed together. The first round is placed in C1, while the second round is placed in C2. This is "Entropy shifting" "Entropy stretching" is achieved by concatenating C1and C2 of rounds 1 and 2 to obtain D1. A repeat of "Entropy shifting" and "Entropy stretching" is done to obtain D2. Entropy mixing is achieved by XORed of D1 with D2. The resultant entropy H, (D.D) is the key K used on TEA shifted to pool 2. The algorithm for entropy shifting is shown in Algorithm 1.

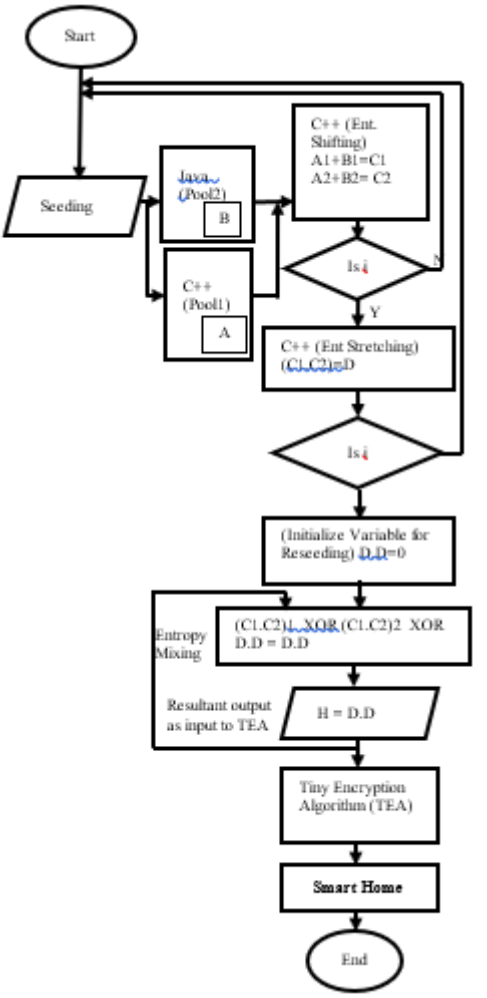

Figure 10. Entropy Generation Technique

Algorithm 1: Entropy-Shifting

Start: Entropy-Shifting

$k < 3$

Seeding

$h_A = \sum_{i=1}^{n} A_1$ /* from pool 1

Call seed function /* from pool 2

For $i = 1$

$C_1 = \sum_{i=1}^{n} A_1 + \sum_{i=1}^{n} B_1$ /*Entropy Shifting (adding B_i to A_i)

Else

$C_2 = \sum_{i=1}^{n} A_2 + \sum_{i=1}^{n} B_2$ /*Entropy Shifting (adding B_i to A_i)

2. *Module 2 - Entropy Stretching*

Pool 2 through a "Call Function" calls the entropy generated in pool 1. Pool 2 and Pool 1 entropies are concatenated (this is stretching). By stretching, the numbers of bits increase from 64bits to 128bits. This increases the time it takes to predict (if possible) the content of the entropy. Algorithm 2 has the algorithm for Entropy Stretching.

Algorithm 2 Entropy-Stretching

$j < 3$

For $j = 1$

$D_1 = (C_1 \bullet C_2)_1$ /* Entropy-stretching (concatenating C_2 to C_1) to have entropy with higher bits)

Else

$D_2 = (C_1 \bullet C_2)_2$ /* Entropy-stretching (concatenating C_2 with C_1) to have entropy with higher bits)

3. *Module 3 – Entropy-Mixing*

The processes: entropy shifting and entropy stretching go through two rounds, and their resulting entropies which are entropy stretching(s) are XORed together to achieve "Entropy Mixing". The algorithm for Entropy Mixing is as shown in Algorithm 3. Table 4 shows the result of a byte.

Table 4 Entropy Mixing – Mixing Bytes

Activity	**Byte**							
Entropy stretched result round 1	1	0	0	0	1	1	0	1
Entropy stretched result round 2	1	0	1	1	0	1	0	0
Final Entropy to be sent to TEA	0	0	1	1	1	0	0	1

Algorithm 3 Entropy-Mixing

Input: $H' \& D$

Output: H

Start: Initialize Reseeding

$R = 0$

For $k < 3$

Call seed function

$H' = D_1 \oplus D_2 \ldots\ldots\ldots\ldots / \oplus = XOR$

$H = H' \oplus R \ldots\ldots\ldots\ldots / reseeding, R$

$R = H$

3.4 Enhanced Tiny Encryption Algorithm (ETEA)

The integration of EGT to TEA solved the problem of related keys and predictable keys attacks in TEA. The EGT produces a new set of four keys every time encryption is to be done. Figure 11 shows the structure of the ETEASH.

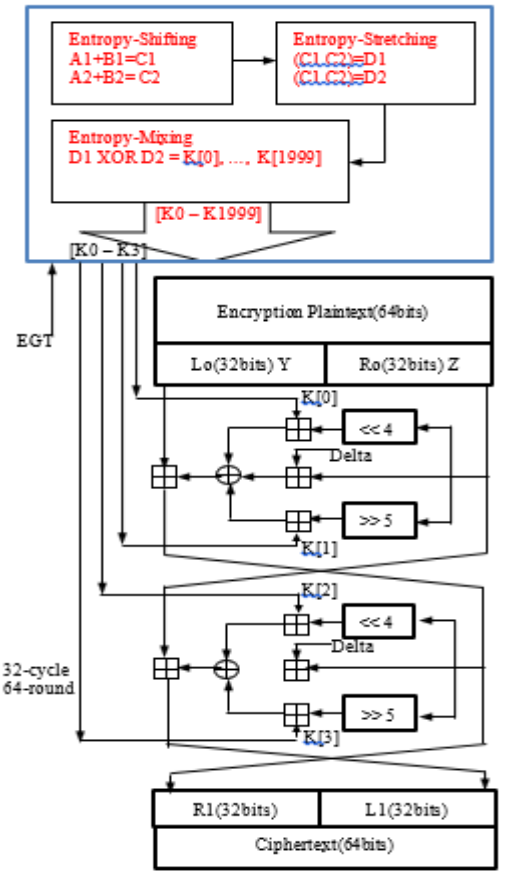

Figure 11. The Structure of the ETEA

The EGT is made up of Entropy shifting, stretching, and mixing. We have two pools of the source of entropies A and B which are our basic sources of EGT creation.

The Formation of Entropy Shifting: The Entropy from pool A called A1 (16-bits) and pool B called B1(16-bits) are or ORed together to give C1. This is repeated and this gives A2 (16-bits) and B2 (16-bits), ORed, giving C2 (16-bits). The resulting bits C1 and C2 are 16-bits.

Entropy shifting function

$$h_A = \sum_{i=1}^{n} A_1 \qquad (1)$$

/* from pool 1

$$h_B = \sum_{i=1}^{n} B_1 \qquad (2)$$

/* from pool 2

$$C_1 = \sum_{i=1}^{n} A_1 + \sum_{i=1}^{n} B_1 \qquad (3)$$

/* Entropy Shifting (adding B_i to A_i) Loops Function 2nd time

$$C_2 = \sum_{i=1}^{n} A_2 + \sum_{i=1}^{n} B_2 \qquad (4)$$

/* 2nd Entropy Shifting (adding B_i to A_i)

The Formation of Entropy Mixing: The inputs to this section of the EGT creation are outputs of Entropy Shifting Formation which are entropies D1 and D2. The key point in this segment of ECG is to strengthen the crypto key by mixing its inputs through the use of XOR acting on them.

$H' = D_1 \oplus D_2 \text{.....} /* \oplus = XOR$

$H = H' \oplus R \text{.....} / reseeding, R.$

$R = H$

This illustration is a sample result of XORing D1 and D2 to give H'

$D_1 = 01000101110101000101010001011101$

$D_2 = 01100110101011101000101000101110$

$H' = 00100011011110101101111001110011$

$H' = 32bitsFinalEntropy$

4. RESULTS

The ETEA was evaluated on the criteria of Entropy level of the EGT using Runs Test, comparison of encryptions of both TEA and ETEASH, and Test of Avalanche Effect to examine its security level.

4.1 Runs Test

The randomness of a set of numbers would be upheld if the set of numbers

do not follow a known or particular sequence and the number of modes is to be minimally acceptable. Hence Runs test was carried out on, the mean, median, mode, and test value 700000000 of the entropy. Our null hypothesis is that the entropy is random. Table 4 to Table 7 show the results.

i. RUNS (MEAN) = Entropy

ii. RUNS (MEDIAN) = Entropy

iii. RUNS (MODE) = Entropy

iv. RUNS (700000000) = Entropy

. Mean: From Table 5, our test value (mean of entropy) is 683962874.91, of a total number of entropy values of 2000. Cases less than the mean are 1204 and cases greater than the mean are 796. The number of runs is 967. The null hypothesis is that our entropy is random if assumed significance level (Asymp. Sig. (2-tailed) is greater than or equal to 0.05.

ii. Median: From Table 6, our test value (median of entropy) is 542508570.00, of the total number of entropy values of 2000. Cases less than median cases are 1000 and cases greater than the median cases are 1000. The number of runs is 1012. The null hypothesis is that our entropy is random if assumed significance level (Asymp. Sig. (2-tailed) is greater than or equal to 0.05

iii. Mode: From Table 7, our test value (mode of entropy) is 1409568548.00, of total number of cases greater than the mean are 796. Number of runs is 967. Null hypothesis is that our entropy is random if assumed significance level (Asymp. Sig. (2-tailed) is greater than or equal to 0.05.

iv. Test Value (700000000.00): From Table 8, our test value is 700000000.00, of total number of entropy values of 2000. Cases less than test value 1204 and

Table 5 Test – Mean

Measuring Parameters	Entropy
Test Value (mean)	683962874.91
Cases < Test Value	1204
Cases ≥ Test Value	796
Total Cases	2000
Number of Runs	967
Z	.36
Asymp. Sig. (2-tailed)	.722

Table 6. Test – Median

Measuring Parameters	Entropy
Test Value (median)	542508570.00
Cases < Test Value	1000
Cases ≥ Test Value	1000
Total Cases	2000
Number of Runs	1012
0Z	.49
Asymp. Sig. (2-tailed)	.623

Table 8 Runs Test – Value (700000000.00)

Measuring Parameters	Entropy
Test Value (mode)	700000000.00
Cases < Test Value	1204
Cases ≥ Test Value	796
Total Cases	2000
Number of Runs	967
Z	.36
Asymp. Sig. (2-tailed)	.722

Table 7 Runs Test – Mode

Measuring Parameters	Entropy
Test Value (mode)	1409568548.00
Cases < Test Value	1997
Cases ≥ Test Value	3
Total Cases	2000
Number of Runs	7
Z	.07
Asymp. Sig. (2-tailed)	.941

4.2 Tiny Encryption Algorithm versus Enhanced Tiny Encryption Algorithm

Table 9 shows encryption with TEA, while Table 10 shows encryption with ETEASH. Figure 12 is an encryption graph comparing TEA's encryption with ETEASH's encryption. Graph of TEA shows that it follows a pattern whose next direction is predictable, implying the "related key" weakness or feature of TEA, while the direction of ETEATH's graph is not predictable implying the strength of the encryption keys. Figure 13, Figure 14 and Figure 15 show the Menu of ETEASH on Raspberry, Encryption and Decryption processes respectively.

Table 9. Encryption with TEA

S/No	Message AB	Message A	Message B	Encryption Keys (K0, K1, K2, K3)	Message A - Encrypted	Message B - Encrypted
1	23248181255	23248	181255	K0=1254665990 K1=1254665000 K2=125460000 K3=1254000000	–1558625664	1563655416
2	23248183255	23248	183255	K0=1254665990 K1=1254665000 K2=125460000 K3=1254000000	–614539806	883765302
3	232481850	23248	1850	K0=1254665990 K1=1254665000 K2=125460000 K3=1254000000	1399674661	–756184618
4	3113155234	3113	155234	K0=1254665990 K1=1254665000 K2=125460000 K3=1254000000	–748546186	124541785
5	3113158239	3113	158239	K0=1254665990 K1=1254665000 K2=125460000 K3=1254000000	1053352235	938921090

Table 10. Encryption with the Developed ETEASH

S/No	Message AB	Message A	Message B	Encryption Keys (K0, K1, K2, K3)	Message A - Encrypted	Message B - Encrypted
1	23248181255	23248	181255	K1=1154665999 K2=15403299 K3=473107876 K4=482607971	1355194468	1859247988
2	23248183255	23248	183255	K1=148404629 K2=1323567688 K3=115604301 K4=1221766670	–1812734587I	–2002224107
3	232481850	23248	1850	K1=1181066263 K2=1131765770 K3=1092875381 K4=1116065613	–329346306	615250651
4	3113155234	3113	155234	K1=86304008 K2=438107526 K3=452307668 K4=1093175384	221171005	123299558
5	3113158239	3113	158239	K1=75403899 K2=97704122 K3=1335367806 K4=1313667589	–1881522647	–1274896088

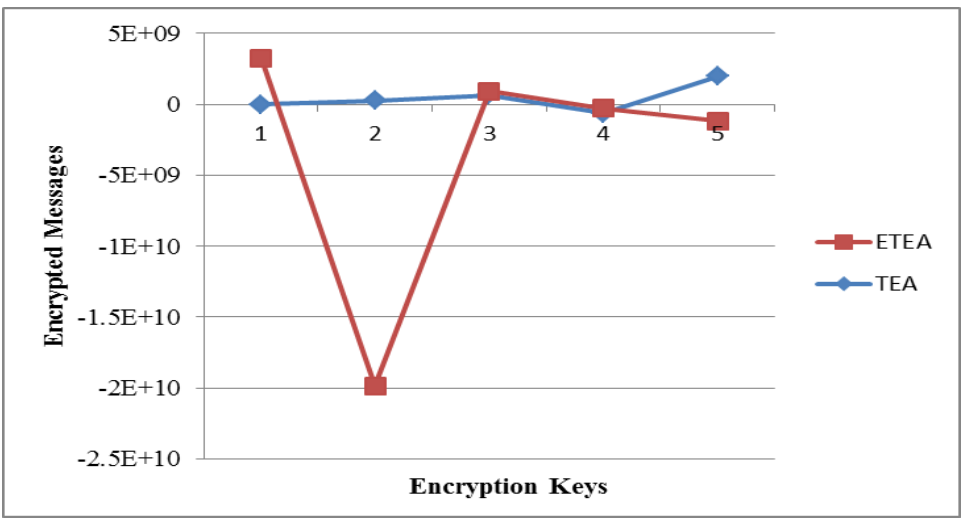

Figure 12.TEA Encryption Keys versus ETEASH Encryption Keys

URL:

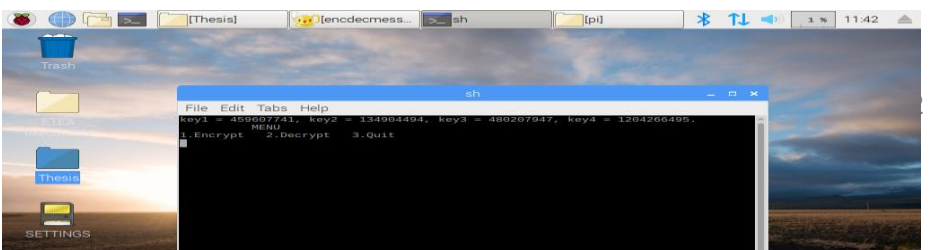

Figure 13. ETEASH Menu on Raspberry Pi3

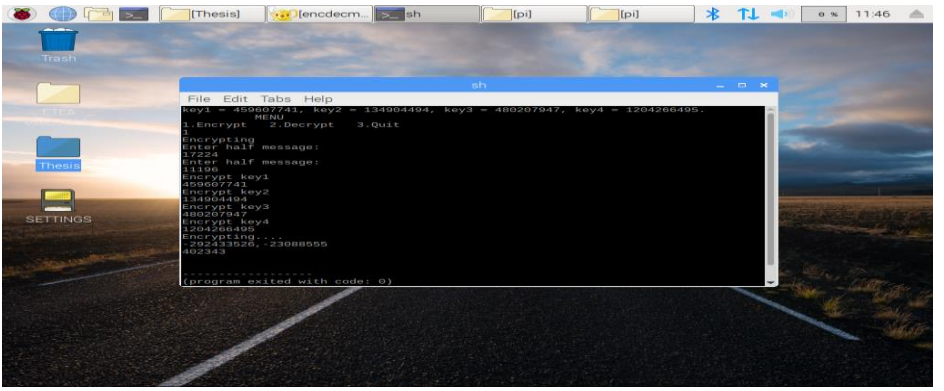

Figure 14. Encryption with ETEASH Encryption Keys

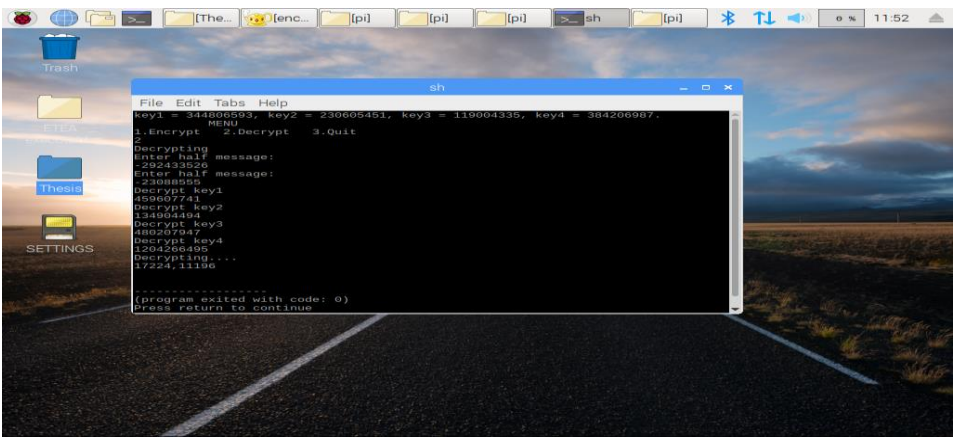

Figure 15. Decryption with ETEASH Encryption Keys

4.3. Test of Avalanche Effect on TEA and ETEASH crypto keys

Avalanche effect in cryptography describes a concept in cryptography where little change in input values leads to a substantial change in the output. The standard TEA was passed through the avalanche test and the Enhanced TEA was also passed through the avalanche Test. Table 11 shows the result of the avalanche effect for TEA's encryption keys while Table 12 shows the avalanche effect for ETEASH encryption keys. Table 13 shows the marginal differences between TEA's encryption keys and ETEASH's encryption keys using the avalanche effect, which ultimately portrays that ETEASH passed the benchmark of Avalanche Test seamlessly compared to TEA. This indicates that the developed ETEASH encryption keys are stronger than TEA's encryption keys. Figure 16 shows the benchmarking of Avalanche Effects of ETEASH and TEA encryption keys, indicating that ETEASH's crypto keys are stronger than TEA's crypto keys.

Table 13 Marginal Differences Between TEA and ETEASH Encryption Keys using Avalanche Effect

S-No. Address	TEA	ETEASH	Marginal Differences
1	48.44	56.25	7.80
2	50.00	57.81	7.81
3	53.13	57.81	4.68
4	60.94	64.06	3.12
5	50.00	56.25	6.25
Ave. Total	262.51	292.18	29.66

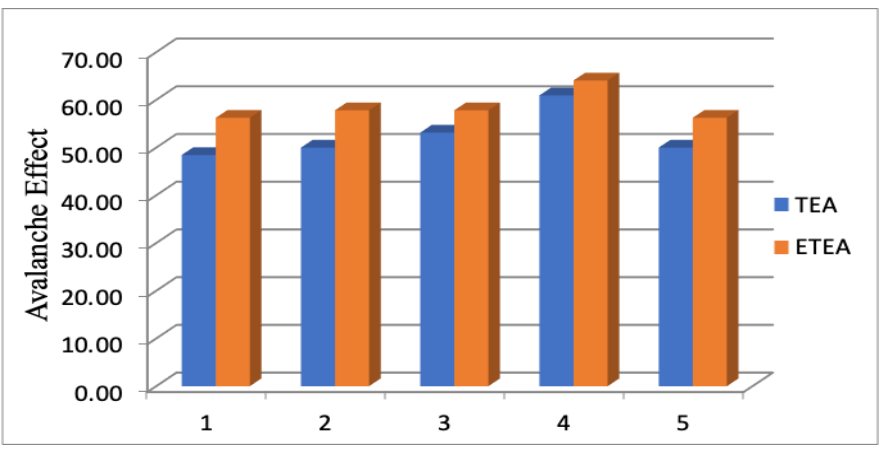

Figure 16. Benchmarking Avalanche Effects of ETEASH Crypto Keys and TEA crypto

4.4. Comparison of Hardware Implementation ETEASH with other

Cryptographic Lightweight Algorithms

A comparison of the hardware implementation of ETEASH with other cryptographic algorithms is shown in Table 14. The parameters for comparison are block size in bits, key size in bits, the code size and RAM size in bytes, and the cycle for encryption and decryption processes

.

Table 14. Comparison of ETEASH with other Cryptographic Algorithms

S/N	Cipher	Device	Block Size (bits)	Key Size (bits)	Code Size (bytes)	RAM (bytes)	Cycle (key+ enc)	Cycles (key+ dec)	Source
1	AES	AVR	64	128	1570	–	2739	3579	(Poettering, 2007)
2	DES	AVR	64	56	4314	–	8633	8154	(Thomas *et al*, 2007)
3	HIGHT	AVR	64	128	5672	–	2964	2964	(Thomas *et al*, 2007)
4	SKIPJACK	Power TOSSIM	64	80	5230	328	17390	–	(Woo *et al*, 2008)
5	RC5	Power TOSSIM	64	128	3288	72	70700	–	(Woo *et al*, 2008)
6	IDEA	AVR	64	80	596	–	2700	15393	(Thomas *et al*, 2012)
7	KATAN	AVR	64	80	338	18	72063	88525	(Thomas *et al*, 2012)
8	mCrypton	AVR	64	96	1076	28	16457	22656	(Thomas *et al*, 2012)
9	KLEIN	AVR	64	80	1268	18	6095	7658	(Thomas *et al*, 2012)
10	PRESENT	AVR	64	128	1000	18	11342	13599	(Thomas *et al*, 2012)
11	PRINCE	AVR	64	128	1574	24	3253	3293	(Aria *et al* 2014)
12	SIT	ATmega328	64	64	826	22	3006	2984	(Muhammad *et al* 2017)
13	TEA	AVR	64	128	648	24	7408	7539	(Thomas *et al*, 2012)
14	**ETEASH**	**Raspberry Pi 3**	**64**	**128**	**3880**	**1000**[3]	**5153**	**4828**	***(This work)***

5. CONCLUSION

IoT provides ease of connectivity and computing capability for everyday communication by ensuring effective data generation and exchange with minimal human intervention. Traditional cryptographic functions for mitigating malicious attacks cannot be applied to low-capacity devices. Therefore, the developed Enhanced TEA on Raspberry pi 3 provides encryption of information and credentials, securing the perception layer of smart home devices against related-key attacks.

The work presented in this paper made the following contributions:

a) The development of a Random Number Generator with Entropy Generation Technique was used in the Enhanced Tiny Encryption Algorithm for constrained devices.

b) The development of Enhanced Tiny Encryption Algorithm for Secured Smart Home.

Future work can be extended to the following areas:

a) Authentication of packets received by a destination node from the source could be carried out, to further strengthen the security of the smart home, in case there is a malicious node within the HAN.

b) This research utilized TEA as lightweight cryptography in securing the smart home with EGT. Other lightweights' cryptographic algorithms could be explored.

c) Other forms of entropy generation and management such as non-deterministic could be explored for better unpredictable key generation and management.

REFERENCES

Abdelhalim, M., El-Mahallawy, M. and A. Elhennawy, M. (2013). Design and Implementation of an Encryption Algorithm for use in RFID System. *International Journal of RFID Security and Cryptography*, 2(1), pp.51-57.

Abdul-Ghani, H. A., Konstantas, D., & Mahyoub, M. (2018). A comprehensive IoT attacks survey based on a building-blocked reference model. *IJACSA) International Journal of Advanced Computer Science and Applications*, 355–373.

Abdulsalam, Y. S., Olaniyi, O. M., & Ahmed,A. (2018). Enhanced tiny encryption algorithm for secure electronic health authentication system. *International Journal of Information Privacy, Security and Integrity*, *3*(3), 230–252.

Abu, M. A., Nordin, S. F., Suboh, M. Z., Yid, M. S. M., & Ramli, A. F. (2018). Design and Development of Home Security Systems based on Internet of Things Via Favoriot Platform. *International Journal of Applied Engineering Research*, *13*(2), 1253–1260.

Ajao, L. A., Kolo, J. G., Adedokun, E. A., Olaniyi, O. M., Inalegwu, O. C., Abolade, S. K. A (2018). Smart door security-based home automation system: An Internet of Things. *Scifed Journal of Telecommunication*, 2(2), 1-9.

Alagbe, V., Popoola S.I., Atayero, A.A., B Adebisi, B. Abolade R.O., Misra S. (2019) Artificial Intelligence Techniques for Electrical Load Forecasting in Smart and Connected Communities.). Lecture Notes in Computer Science, vol.

11623 (pp. 219-230). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-24308-1_18

Alharbi, A., Zohdy, M., Debnath, D., Olawoyin, R., & Corser, G. (2018). Sybil Attacks and Defenses in Internet of Things and Mobile Social Networks. *International Journal of Computer Science Issues (IJCSI)*, *15*(6), 36–41.

Ali, I., Sabir, S., & Ullah, Z. (2016). *Internet of Things Security, Device Authentication and Access Control: A Review*. *14*(8), 11.

Alizadeh M., Shayan, J., Zamani, M., & Khodadadi, T. (2012). *Code analysis of lightweight encryption algorithms using in RFID systems to improve cipher performance.* Paper presented at the Open Systems (ICOS), 2012 IEEE Conference.

Aria S., Cong C. & Thomas E. (2014). AVRprince – AnEfficient Implementation of PRINCE for 8-bit Microprocessors. Worcester Polytechnic Institute, Worcester, MA, USA {ashahverdi,cchen3,teisenbarth}@wpi.edu

Bhosale, S. D., & Sonavane, S. S. (2018). Wormhole attack detection in internet of things. *International Journal of Engineering & Technology, 7 (2.33) (2018) 749-751*. Website: www.sciencepubco.com/index.php/IJET

Buenrostro, E., Cyrus, D., Le, T., & Emamian, V. (2018). Security of IoT Devices. *Journal of Cyber Security Technology*, *2*(1), 1–13.

Burhan, M., Rehman, R. A., Khan, B., & Kim, B.-S. (2018). IoT elements, layered architectures and security issues: A comprehensive survey. *Sensors*, *18*(9), 2796.

Das, S. R., Chita, S., Peterson, N., Shirazi, B. A., & Bhadkamkar, M. (2011). Home automation and security for mobile devices. *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 141–146.

De Leon, R. M., Sison, A. M., & Medina, R. P. (2019). A Modified Tiny Encryption Algorithm Using Key Rotation to Enhance Data Security for Internet of Things. *2019 International Conference on Information and Communications Technology (ICOIACT)*, 56–60.

ElShafee, A., & Hamed, K. A. (2012). Design and implementation of a WIFI based home automation system. *World Academy of Science, Engineering and Technology*, *68*, 2177–2180.

Hasan, R., Khan, M. M., Ashek, A., & Rumpa, I. J. (2015). Microcontroller Based Home Security System with GSM Technology. *Open Journal of Safety Science and Technology*, *5*(02), 55.

Hossain, M. K., Biswas, P., Mynuddin, M., & Morsalin, S. (2014). Design and implementation of smart home security system. *International Journal of Modern Embedded System*, *2*(6), 7–10.

Hughes, R., & Nordholt, J. (2016). *Strengthening the security foundation of cryptography with Whitewood's quantum-powered entropy engine*. January.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). Security of the Internet of

Things: Perspectives and challenges. *Wireless Networks*, *20*(8), 2481–2501.

Jordi M. B., Athanasios V., & Mariusz G. (2017). Secure Smart Homes: Opportunities and Challenges. ACM Comput.Surv. 50, 5, Article 75 (September 2017), 32 pages.https://doi.org/10.1145/3122816

Karimi, K., & Krit, S. (2019). Smart home-Smartphone Systems: Threats, Security Requirements and Open research Challenges. *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 1–5.

Kim, T. H.-J., Bauer, L., Newsome, J., Perrig, A., & Walker, J. (2011). Access right assignment mechanisms for secure home networks. *Journal of Communications and Networks*, *13*(2), 175–186.

Kumar, K. V., Mascarenhas, S. J., Kumar, S., Rakesh, J. P. V., & Kumar, V. K. (2015). Design and implementation of Tiny encryption algorithm. *International Journal of Engineering Research and Applications, ISSN*, 2248–9622.

Kumar, R., & Mittal, P. (2019). A Novel Design and Implementation of Smart Home Security System: Future Perspective. *International Journal of Applied Engineering Research*, *14*(2), 363–368.

Ling, Z., Luo, J., Xu, Y., Gao, C., Wu, K., & Fu, X. (2017). Security vulnerabilities of internet of things: A case study of the smart plug system. *IEEE Internet of Things Journal*, *4*(6), 1899–1909.

Misra, S. A Step by Step Guide for Choosing Project Topics and Writing Research Papers in ICT Related Disciplines. In *Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24–27, 2020, Revised Selected Papers 3* (pp. 727-744). Springer International Publishing

Muhammad U, Irfan A, M, Imran A, & Shujaat K, & Usman A, S. (2017) SIT: A Lightweight Encryption Algorithm for Secure Internet of Things. International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1,

Oluwade O. R., Olaniyi O. M., Abdulsalam Y. S. & Ajao L.A (2018). Entropy Generation Technique In Lightweight Cryptographically Secured Smart Home. 12th International Multi-Conference on Information Communication Technology Applications (AICTTRA, Obafemi Awolowo University)

Oluwu M., Yinka-Banjo C., Misra S., Oluranti J. (2020). Internet of Things: Demystifying Smart Cities and Communities. Lecture Notes in Networks and Systems, 2020, 119, pp. 363–371 Springer, Singaporehttps://rd.springer.com/chapter/10.1007/978-981-15-3338-9_41

Pavithra, M. (2019). *IOT Home Automation with PIR Sensor Security using ESP8266 WI-FI Chip and GSM*. *8*(6), 5.

Poettering, B. (2007). Rijndael furious aes-128 implementation for avr devices. Retrieved from: http://point-at-infinity.org/avraes/rijndaelfurious.asm, on 03/04/2020.

Popoola, S. I., Popoola, O. A., AI Oluwaranti, A. I. Atayero A.A., Badejo J.A., Misra S. (2018). A Cloud-Based Intelligent Toll Collection System for Smart Cities. Communications in Computer and Information Science book series (CCIS, volume 827 pp. 653-663). Springer, Singapore. https://rd.springer.com/chapter/10.1007/978-981-10-8657-1_50

Raggett, D. (2016). *Tackling Data Security and Privacy Challenges for the Internet of Things*. Retrieved from W3C: https://www. w3. org/Talks/2016/0614-iot-security. pdf.

Rambus. (2019). Smart Home: Threats and Countermeasures. Retrieved from Retrieved On March 29, 2020 from https://www.rambus.com/iot/smart-home/

Rao, T. A. (2018). Security challenges facing IoT layers and its protective measures. *International Journal of Computer Applications*, *975*, 8887.

Shaikh, E., Mohiuddin, I., & Manzoor, A. (2019). Internet of Things (IoT): Security and Privacy Threats. *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, 1–6.

Shifa, A., Asghar, M. N., Batool, N., & Fluery, M. (2019). Efficient Lightweight Encryption Algorithm for Smart Video Applications. *ArXiv Preprint ArXiv:1901.08344*.

Shoeb, M., & Gupta, V. K. (2013). A crypt analysis of the tiny encryption algorithm in key generation. *International Journal of Communication and Computer Technologies*, *1*(38).

Souza L S, Misra S, & Soares M, S. (2020) SmartCitySysML: A SysML Profile for Smart Cities Applications. Lecture Notes in Computer Science, 12254 LNCS, pp. 383–397 Springer, Cham. https://rd.springer.com/chapter/10.1007/978-3-030-58817-5_29

Suchitra, C., & Vandana, C. P. (2016). Internet of things and security issues. *International Journal of Computer Science and Mobile Computing*, *5*(1), 133–139.

Thomas E., Christof P., Axel P., Sandeep K., and Leif U. (2007). A Survey of LightweightCryptography Implementations. Design and Test of ICs for Secure Embedded Computing. IEEE Design & Test of Computers.

Thomas E., Zheng G., Stefan H., Sebastiaan I., Tomislav N., Thomas P., and Francesco R. (2012). Compact Implementation and Performance Evaluation of Block Ciphers in A Tiny Devices. International Conference on Cryptology in Africa.

Usman, M., Ahmed, I., Aslam, M. I., Khan, S., & Shah, U. A. (2017). SIT: A lightweight encryption algorithm for secure internet of things. *ArXiv Preprint ArXiv:1704.08688*.

Vassilev, A. T., & Hall, T. (2014). The Importance of Entropy to Information Security. *Computer (IEEE Computer)*, *47*(2). https://www.nist.gov/publications/importance-entropy-information-security

Woo K. K., Hwaseong L., Yong H. K., and Dong H. L. (2008) Implementation and Analysis of Lightweight Cryptographic

Algorithm suitable for Wireless Sensor Networks. Information Security and Assurance, 2008. ISA 2008. International Conference on IEEE.

Yassein, M. B., Hmeidi, I., Shatnawi, F., Mardini, W., & Khamayseh, Y. (2019). Smart Home Is Not Smart Enough to Protect You-Protocols, Challenges and Open Issues. *Procedia Computer Science*, *160*, 134–141.

Zandamela, A. A. (2017). An Approach to Smart Home Security System using Arduino. *Electrical Engineering: An International Journal (EEIJ)*, *4*(2/3), 1–18.

Zeadally, S., Das, A. K., & Sklavos, N. (2019). Cryptographic technologies and protocol standards for Internet of Things. *Internet of Things*, 100075.

Zhang, J., Duong, T. Q., Woods, R., & Marshall, A. (2017). Securing wireless communications of the internet of things from the physical layer, an overview. *Entropy*, *19*(8), 420.