EAI Endorsed Transactions
on Security and Safety

Research Article **EAI.EU**

Development of a Multifactor-Security-Protocol System Using Ambient Noise Synthesis

Agbotiname Lucky Imoize[1,*], Boluwatife Samuel Ben-Adeola[1], John Adetunji Adebisi[1]

[1] Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Lagos, Akoka, Lagos, Nigeria

Abstract

The escalating cases of security threats on the global scene, especially in the cyberspace, demands urgent need to deploy sophisticated measures to mitigate these calamitous threats. To this end, various lock mechanisms have been developed and deployed to prevent access to control systems from potential intruders. This paper provides a solution to this pervasive problem, addressing concerns on the physical and virtual components of an access control system. A locally generated One-Time-Passkeys (OTPs) was created, leveraging ambient noise as entropy input. The system was deployed on an Arduino microcontroller embedded in a safe-cabinet secured with a 12V solenoid lock. The design was implemented and tested against standard metrics. Results achieved include algorithmic optimizations of existing local OTP protocol implementations, and the realization of a safe lock module, which interfaces with a mobile application developed on Android over a secured Bluetooth connection.

Keywords: Multifactor-Authentication, One-Time-Passkey (OTP), Security systems, Bluetooth connectivity, Sound-synthesis, Cyber security, Internet of Things (IoT).

Received on 25 January 2020, accepted on 10 April 2020, published on 15 April 2020

doi: 10.4108/eai.13-7-2018.163979

1. Introduction

The development of advanced technological solutions that are capable of protecting highly sensitive systems from malicious threats is undoubtedly one of the greatest needs of mankind in recent times [1]. Generally, security does not involve considerations of the mechanical strength of systems only, but also that of the cyber, virtual, and systems infrastructure. With the current rise in the application of internet of things (IoT) [2] [3], which seeks to make all domestic and commercial devices internet-connected, it is imperative that security solutions strive to minimize dependence on the existing penetrable internet solutions and instead, deploy local and offline components in the design of functional and robust security systems [4]. Toward this end, this paper addresses security concerns through the design of a multi-factor authentication system leveraging local ambient noise for one-time passkey (OTP) generation.

Several mechanisms have been reported to defend systems against cyber-attacks, especially for well-defined applications. The local OTP generation has been adopted in the current undertaking. However, the implementation intricacies for random sequence generation using ambient noise levels could results in limited system performance under certain conditions. This performance is measured by varying measures of entropy and randomness such as birthday spacings, overlapping permutations, and random spheres test. The conditions under which the performance of the implementation adopted begin to underperform on the previously stated benchmarks includes: low noise environment with little atmospheric disturbance, low noise-variance environments even when the amplitude of disturbance is high. The extent of the incorrectness or errors with this implementation under the given conditions is considered and significantly improved in the proposed multifactor-security-protocol system leveraging ambient noise synthesis.

[*] Corresponding author. Email: aimoize@unilag.edu.ng

The proposed system design is being deployed on an Arduino microcontroller embedded in a safe-cabinet secured with a 12V solenoid lock. To ensure user-friendliness of the system, various components including an android mobile application was developed for interfacing with the lock module. To ascertain optimization results of the system implementation, numerous tests were conducted in various uncontrolled environments and benchmarked against standard metrics, and the results obtained were compared with the performances of related solutions. Finally, the evolved system was demonstrably shown to outperform existing models in terms of true-randomness of secure OTP keys.

Restriction of access and protection of property is ubiquitous [5]. However, most solutions available to consumers, use manual mechanisms with various limitations, and most cutting-edge solutions are limited to the industrial environments due partly to high cost of deployment and implementation [6]. The design outcome brings this cutting-edge solution within the reach of the consumers, leveraging the ubiquitous smartphone for interfacing with the hardware to help reduce the cost, and also bypasses the need for an expensive computational platform. The development is seen to be versatile and durable enough to be used in certain commercial applications and environments, thus bringing the financial benefits of affordability and the convenience offered by its mobility, especially when deployed in information and communication technology (ICT) – dependent business environments [7][8].

Last, the design implements an electromagnetic lock mechanism with logic control from an Arduino microcontroller, which interfaces with an Android mobile application over a secure Bluetooth connection. The security of the system is based on a locally generated one-time-password (OTP) protocol, which leverages ambient room noise for generation of random number stream, making it difficult for interference from malicious third-parties over internet connections.

The remaining part of this paper is organized as follows. Section 2 presents related work and theoretical background. Section 3 focuses on the design methodology and specifications. Section 4 presents the results and discussions, and finally, the conclusion to the paper and future perspectives are given in Section 5.

2. Related Work

In the open literature, several papers have been reported on security systems with different implementation techniques. Most of the implementations could be described according to their categorization along each dimension as shown in Fig. 1. This comprises of the lock mechanism that is made up of the electrical and mechanical components, the key type comprising of the unique and uniform lock system architectures, and the authentication type made up of the single and double factor authentications. Various authentication mechanisms are surveyed by Meng *et al* [6], the concept of Single Factor Authentication (SFA) is given by Potts and Sukittanon [11], and the Double Factor Authentication (DFA) was reported by Kamelia *et al* [12]. Furthermore, Jain, Shukla, and Rajan [14] reported a uniform key approach to the said authentication problem, which appears to be a less secure system. For an enhanced security of an authentication system, Sharma *et al* [15] utilized Global Positioning System (GPS) technology along with Global System for Mobile Communications (GSM) and Bluetooth technologies to provide additional layers of authentication, and security for the locking mechanism of a motor vehicle door.

In a related study, Suryawanshi *et al* [16] reported an implementation for the busy and disabled people leveraging both a secret PIN input over a secure Wi-Fi connection reinforced by (or used alternatively) with voice activation using speech recognition software. In the same vein, Trisnani *et al* [17] developed a two-user subclass for common and administrative users. The common users have authority to lock and unlock secured doors. The admin users have authority to access secured doors and view activity logs on who has had access to the door, and Navya and Ramachandran [18] leveraged on social media platforms such as Facebook and WhatsApp to send logs to users when someone is granted access to the lock. The direct messaging features on these platforms could also be used for sending one-time-passwords OTPs for implementations where they are utilised.

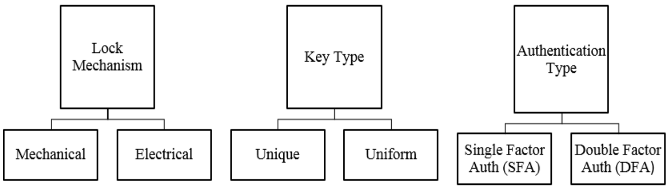


Fig. 1. Dimensions of analysis of the Lock System Architectures.

Recently, Guo *et al* [9] reported a mechanical lock mechanism, which involves moving a physical barrier via electrical excitation of an electronic servo-motor. The merits of this system include the reliability of the lock system in the case of power failure, albeit, only when it is inside the contained unit (i.e. when inside the locked room or vessel). This system, however, may be subject to some structural defects as a result of the relatively weak materials used in the fabrication of the servo-mechanism.

In contrast to the design presented by Guo *et al*, the implementation by Ismail [10] uses an electromagnetic lock mechanism. This module is excited when electrical current flows through the solenoid embedded in the module, engaging the lock and securing the unit of interest. The pros of this system include relative invulnerability to forced entry as compared to the servo system proposed by Guo *et al* [9]. However, the system is not readily reliable in the case of power outage, and this is seen as a major drawback. Interestingly, it is worthy of note that there are two variants of this module, which offer different lock behaviours in the case of the absence of excitation such as power failure.

On the last dimension of analysis for the system architecture key-type, the implementation reported by Kumar *et al* [13] is reviewed. In this work, the key-type implemented was quite unique. By this, the nature of the secret key is assigned to users in the system. The major question under consideration of the nature of the key is whether it is unique to each user if the same key is uniformly assigned. As seen in the report, each private-public key pair is unique to every user. That is, along with unique token given to each installation of the app, the password (secret/private key) generated alongside is unique to that key, therefore each user can be uniquely identified by the pair of keys used in gaining access.

Shin, Han, and Jin [19] improved on the work in [18] using One-time passwords (OTPs) for authentication. The merit of the design reported by Shin *et al* [19] is that a common user attempting to access the lock does not need the help of the administrative user, given the OTP process is known to the user, and the user possesses an authorised and reregistered phone number. In this system, the OTP would come in over an SMS or other messaging or voice channels such as a phone call or social media platforms as described in [18], the user then inputs the received OTP into the system, and is granted access accordingly. As a fall-back mechanism for when the user’s phone is not available or off at the time of request for the OTP, then a security question will be presented to the user for the account which is used to request access. This security question is pre-set during the on-boarding process of account creation.

The problem addressed by Prada-Delago *et al* [20] is the security vulnerability of storing secret passkeys in remote servers. This problem is solved by bypassing the need to store keys beforehand, rather, this security is achieved by generating the required keys on-demand, that is when they are needed and doing away with the keys when they are not useful. These passkeys are generated/constructed using physical tokens which are possessed by the authorised users and utilize the start-up values in the Static random-access memory (SRAM) of the Bluetooth Low Energy (BLE) chip of the token. This has added security advantage as this key construction performs the function of a physical unclonable function (PUF), meaning in the potential case of the theft of the generated key, the lock remains well secured.

Furthermore, Hsu *et al* [21] reported that the greatest motivation for designing security systems is affordability and cost efficiency in the proposed product, and its prototype is to serve primarily the low-income earners in far-east Asia. This influenced the decision in the fabrication materials used and communication modules implemented as well. An example of how it influenced the design considerations, is the decision to use Bluetooth Low Energy which would consume less power and be less of a burden to the proposed owners in terms of power consumption. The device supported voice recognition as well, however this feature was supported by the mobile application which requires an interface with the lock; another design consideration to cut cost, as this decision allowed them to by-pass the need for a standalone microphone, which would have added to the cost of the lock.

Anitha [22] explored the QR code technique for system authentication. To gain access to the lock, the user scans the QR code sent to their registered device from a remote server with the details encrypted using the MD5 encryption algorithm and stored on the remote database as a checksum. When access is requested the scanned code is sent to the remote server and is compared to a valid checksum if there is a match, access is granted, otherwise the user is denied access. In Dabhade *et al* [23], a lock mechanism was reinforced with an alarm system which involves sending a photograph taken from the camera installed in the unit at the time of the alarm trigger to the email address of the owner of the home. The alarm system also has a buzzer unit which sounds a very audible alarm to alert people in the vicinity of the attempted intrusion. After a comprehensive review of the aforementioned implementations, some decisions were made critical for the proposed Lock System.

Another report that is closely related to the lock system proposed in this paper is the work of Regis [24], which is more concerned about malicious third-party interference with internet of things (IoT) security protocols. Many of such IoT applications are vital to the operation of businesses and Government parastatals, thereby making security at all levels of utmost importance. The design decision was made to by-pass the need for remote one-time password (OTP) generation and to instead have the system leverage local ambient noise for random sequence generation. With this implementation, there is reduced vulnerability of the system to man-in-the-middle attacks, which involve malicious parties interfering with the transfer stream of the one-time password, thereby compromising the channel’s integrity.

In the current implementation, the microcontrollers used come programmed with a bootloader which simplify uploading of software programs to the on-chip flash. On the Arduino UNO, the Opti bootloader is utilized [25]–[27]. Also, Bluetooth is used in the exchange of information between the associated devices [9],[23]. This is because Bluetooth utilizes a radio transmission technology called Frequency-hopping Spread Spectrum

(FHSS), and this enables the separation of sent information into chunks known as "packets". These packets are then transmitted individually on one of the seventy-nine Bluetooth communication channels [28]. The module operates at an enhanced data rate of 3Mbps Modulation [29]–[31] with 2.4GHz radio transceiver and baseband, and the device operates with a Master-Slave architecture. Finally, an exchange of keys then takes place, given there is a successful exchange, the pairing is complete as illustrated in Fig. 2. In addition, Multi-factor authentication (MFA) is employed in our design as shown in Fig. 3. Two-factor authentication (2FA) is a subset of multi-factor authentication (MFA) [32]. An illustrative scenario applied in this study is a system where to supplement a user-controlled password, a one-time password (OTP) or code generated or received by the authorized user requesting access, and this follows the idea in [33].

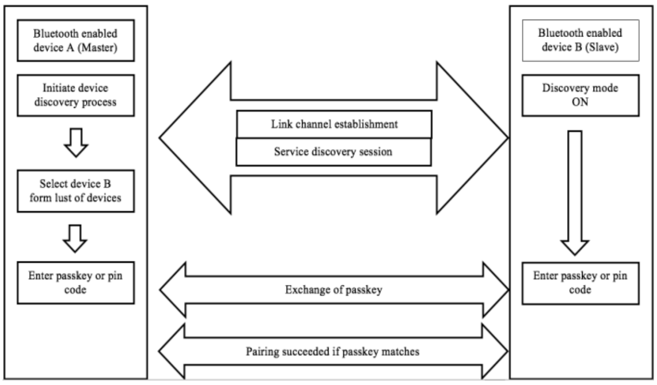

Bluetooth enabled device A (Master)
Initiate device discovery process
Select device B form lust of devices
Enter passkey or pin code
Link channel establishment
Service discovery session
Exchange of passkey
Pairing succeeded if passkey matches
Bluetooth enabled device B (Slave)
Discovery mode ON
Enter passkey or pin code

Fig. 2. The Bluetooth pairing process with master-slave architecture.

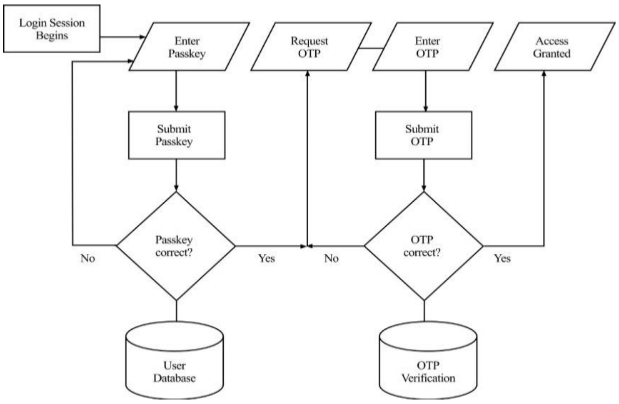
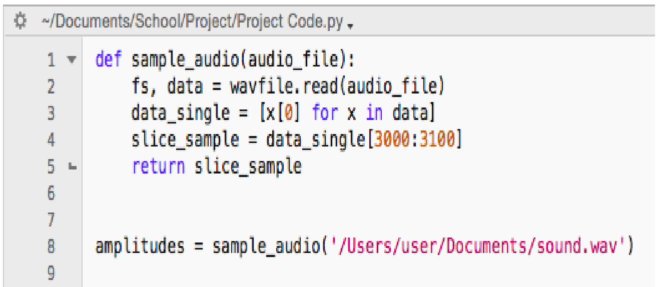


Fig. 3. Multifactor authentication flowchart

3. Design Methodology

3.1 Software Design of Password Generation

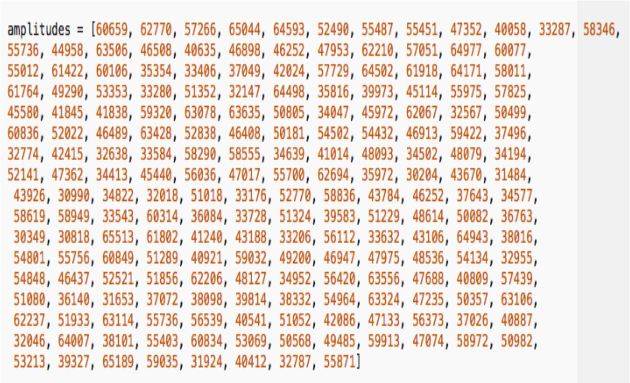
Various algorithms and programming code were used for the development of the audio sampling components of local one-time-password OTP generation. The algorithm was primarily implemented in Python while the microcontroller code was written in C++. Towards the generation of the random OTP from ambient sound, the audio recording was first sampled for amplitude at 44,100 sample rate (same as it was recorded in) using the python “wavfile” library. This library take each sample in the audio recording and casts each amplitude level to the same decimal representation of the bit resolution it was recorded in. Second, the input file was recorded with a 16-bit recorder, hence the range of acceptable values is 0 to 65,535(2^{16}-1). For the sake of memory considerations, only the first 100 samples – 3000 to 3100 – were obtained and stored for further processing, as shown in Figs. 4 and 5, respectively.



```
def sample_audio(audio_file):
    fs, data = wavfile.read(audio_file)
    data_single = [x[0] for x in data]
    slice_sample = data_single[3000:3100]
    return slice_sample


amplitudes = sample_audio('/Users/user/Documents/sound.wav')
```

Fig. 4. Python code snippet for audio sampling



```
amplitudes = [60659, 62770, 57266, 65044, 64593, 52490, 55487, 55451, 47352, 40058, 33287, 58346,
55736, 44958, 63506, 46508, 40635, 46898, 46252, 47953, 62210, 57051, 64977, 60077,
55012, 61422, 60106, 35354, 33406, 37049, 42024, 57729, 64502, 61918, 64171, 58011,
61764, 49290, 53353, 33280, 51352, 32147, 64498, 35816, 39973, 45114, 55975, 57825,
45580, 41845, 41838, 59320, 63078, 63635, 50805, 34047, 45972, 62067, 32567, 50499,
60836, 52022, 46489, 63428, 52838, 46408, 50181, 54502, 54432, 46913, 59422, 37496,
32774, 42415, 32638, 33584, 58290, 58555, 34639, 41014, 48093, 34502, 48079, 34194,
52141, 47362, 34413, 45440, 56036, 47017, 55700, 62694, 35972, 30204, 43670, 31484,
 43926, 30990, 34822, 32018, 51018, 33176, 52770, 58836, 43784, 46252, 37643, 34577,
 58619, 58949, 33543, 60314, 36084, 33728, 51324, 39583, 51229, 48614, 50082, 36763,
 30349, 30818, 65513, 61802, 41240, 43188, 33206, 56112, 33632, 43106, 64943, 38016,
 54801, 55756, 60849, 51289, 40921, 59032, 49200, 46947, 47975, 48536, 54134, 32955,
 54848, 46437, 52521, 51856, 62206, 48127, 34952, 56420, 63556, 47688, 40809, 57439,
 51080, 36140, 31653, 37072, 38098, 39814, 38332, 54964, 63324, 47235, 50357, 63106,
 62237, 51933, 63114, 55736, 56539, 40541, 51052, 42086, 47133, 56373, 37026, 40887,
 32046, 64007, 38101, 55403, 60834, 53069, 50568, 49485, 59913, 47074, 58972, 50982,
 53213, 39327, 65189, 59035, 31924, 40412, 32787, 55871]
```

Fig. 5. Array of 100 audio samples, output from wavfile library function

3.2 Optimisation algorithm

Following the capturing of 100 sample values obtained from the audio file, there is an intermediate optimisation applied to the values. This optimisation is to ensure there is enough requisite entropy for the generation of the random one-time-password OTP. This algorithm works by ensuring contiguous samples in a stream are sufficiently varied, which may cause problems of predictability in the case of a room with little to no audio turbulence. The “thresh” variable in the algorithm represents the required minimum difference between the amplitude values of any given two contiguous samples in a stream as illustrated in Fig. 6, and the snippet for the algorithm optimisation is given in Fig. 7. It has currently been set to 5000. The “target” variable represents the required number of sufficiently varied samples needed for the next step; the hashing of values for the OTP generation. When the samples target has been reached, the collected samples are returned as an array, and the array is stored in a variable “seeds.”

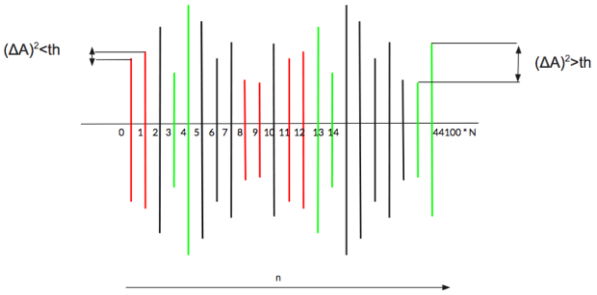
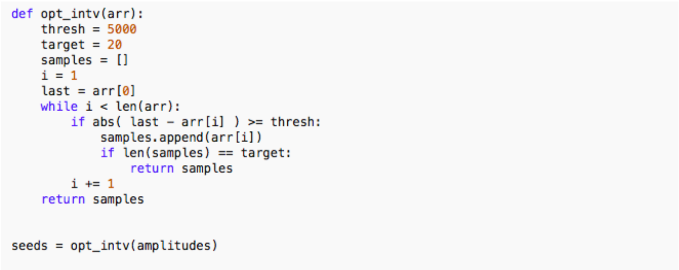


Fig. 6. Variation of contiguous samples in a stream for the algorithm optimisation



```
def opt_intv(arr):
    thresh = 5000
    target = 20
    samples = []
    i = 1
    last = arr[0]
    while i < len(arr):
        if abs( last - arr[i] ) >= thresh:
            samples.append(arr[i])
            if len(samples) == target:
                return samples
        i += 1
    return samples


seeds = opt_intv(amplitudes)
```

Fig. 7. Sample snippet for the algorithm optimisation

3.3 Android mobile application

An android mobile application which serves as the interface between users and the hardware lock unit embedded in the door was developed. The medium of communication between the android application and the lock unit is a secure 2.4GHz-channel Bluetooth connection. The software application requires; text input field for entering secret keys, automatic Bluetooth pair initiation with previously paired lock, Request for OTP over a secure HTTPS internet connection to a remote server and Receive SMS messages containing the OTP as received from the remote server. Fig. 8 shows the wireframes for the design layout of the mobile application used in our proposed scheme.

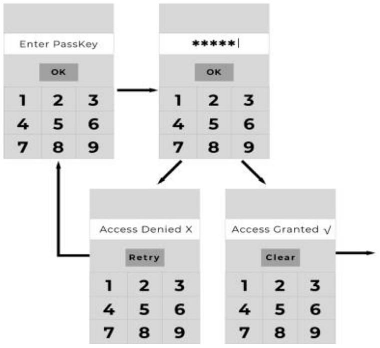


Fig. 8. Wireframes for Android mobile application to be used with lock system

3.4 Hardware Subsystem

The hardware components used for the implementation include: ATmega328 (Arduino Uno), Electromagnetic Solenoid Lock, HC-05 Bluetooth Module, 20MHz Capacitors, 5V Voltage regulators, 16 X 1 LCD display screen, TIP120 Transistor, 1N4001 Diode, Electrolytic Capacitors and 7805 5V voltage regulator. The power supply requirements of the entire system by examining each of its component requirements using suitable specifications for the entire system through simple mathematical analysis is as follows:

a. **Atmega328 (Arduino Uno)**
Operating Voltage V_{CC} = 5V
Current Consumption at 25°C = 5.2mA
Power dissipation = 5 × 5.2 × 10^{-3}= 26mW

b. **16 X 1 Liquid Crystal Display**
Maximum input voltage = 5V
Operating voltage = 3.3V
Maximum current consumption = 25mA
Power dissipation =
3.3 × 25 × 10^{-3}=82.5mW

c. **Electromagnetic Solenoid Lock**
Maximum input voltage = 12V
Operating voltage = 9V
Maximum current consumption = 650mA
Power dissipation = 9 × 650 ×10^{-3} = 5.85W

It is worthy to note that the current requirements for the solenoid lock surpass that which the Arduino board can safely provide, an external 12v source will be needed to supply this current. This source will be the same that powers the Arduino, but before it is stepped down to 5V.

d. **HC-05 Bluetooth Module**
Maximum input voltage = 6V
Operating voltage = 3.3V
Maximum current consumption = 150mA
Power dissipation = 3.3 × 150 ×10^{-3}= 495mW

For the electromagnetic solenoid lock, current only flows when activated for unlock, that is there is only power dissipation when the lock is activated to unlock. Therefore, we considered two scenarios for total power dissipation by the system. First, when lock is activated, and when lock is in idle state. Total power dissipation when lock is activated is given as:
26mW + 82.5mW + 5850mW + 495mW = 6453.5mW = 6.45W
Under idle state, total power is given as: 26mW + 82.5mW + 495mW = 603.5mW.

By inspection, it is observed that, the discrepancy in the total power dissipation for both scenarios only differs by a value of 5850mW, which is the power drawn during the lock’s activation, with its power drawn in idle state being negligible. The end-to-end block diagram is as shown in Fig. 9. The flow diagram of the complete system implementation is given in Fig. 10. In this implementation, the system is started and the Bluetooth pairing process is initiated followed by entering the passkey. If the passkey is not correct, a prompt request is issued to enter passkey again. However, if the passkey is correct, OTP is requested and if the OTP is not valid, there would be need to enter the correct OTP to disengage the lock, and a 5 seconds delay is allowed before the lock is re-engaged and the process is completed.

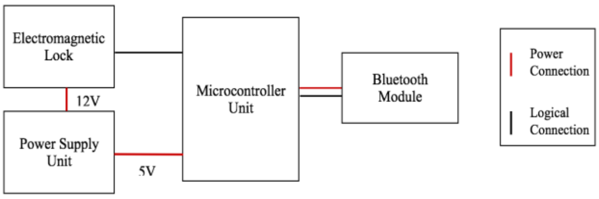


Fig. 9. Block Diagram of the Hardware Subsystem showing end-end implementation

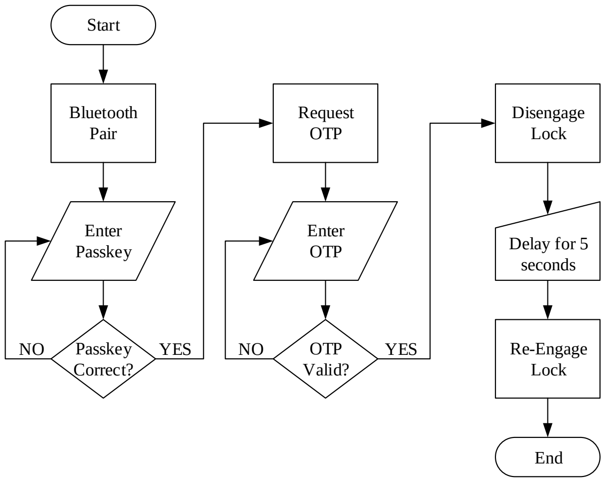


Fig. 10. Flow Diagram of the system implementation

4. Results and Discussion

This study developed and implemented an electromagnetic lock mechanism with logic control from an Arduino microcontroller, which interface with a designed Android mobile application over a secure Bluetooth connection. The android application interface is as shown in Fig. 11. The application was designed such that the "Request OTP" user interface button is clicked, a sound clip is recorded and synthesized according to predefined algorithms, the output is then passed as an output one-time password OTP which is sent to the user over the secure Bluetooth connection and can be passed for access request.

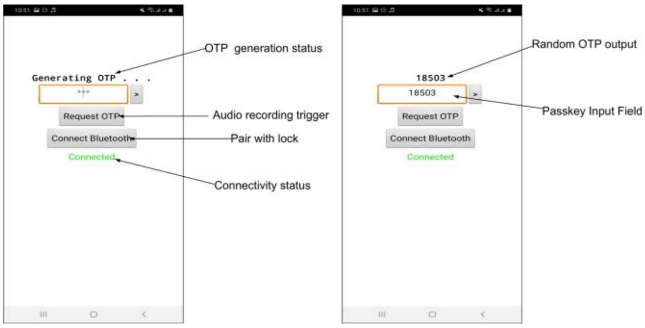


Fig. 11. The Android Application Interface showing request OTP

The local one-time password (OTP) generator was implemented along with algorithmic optimisations providing improved performance of the random sequence generator by introducing enough entropy to the Arduino Rand() function needed for the OTP generation. The performance of the local OTP generation under pre and post-optimisation conditions was evaluated using a scatter plot graph of randomly generated sequences as output from the Arduino's Rand() Function. The plot is three dimensional represented by the colour coding intensity. The third dimension is "frequency of recurrence." What this means is that we are judging a random sequence generator by its ability to generate unique number sequences after multiple iterations. If all numbers in one sequence are repeated in subsequent generation iterations, then this represents a predictable and thus an ineffective underlying sequence generation engine. The total number of iterations was chosen as 100, and the range of numbers set at 1000 to 10000 (9000 possible values) and a sequence length of 100 was chosen. The results with and without the audio seed pre-optimization and post-optimization coupled with other output considerations are depicted in Figs. 12–22.

Specifically, Fig. 12 gives the random number generation output frequency scatter plot without audio seed, Fig. 13 depicts the random number generation output frequency scatter plot with audio seed, pre-optimisation with noise, whereas, Fig. 14 presents the random number generation output frequency scatter plot with audio seed, pre-optimisation with no noise, and Fig. 15 gives the random number generation output frequency scatter plot with audio seed, post-optimisation with no noise. In addition, Fig. 16 displays the no-seed pre-optimisation high noise scenario, Fig. 17 gves the no-seed pre-optimisation low noise scenario, Fig. 18 presents the plot with-seed pre-optimisation high noise scenario, Fig. 19 gives the plot with-seed pre optimisation low noise scenario I. Similarly, Fig. 20 shows a plot with-seed pre-optimisation low noise scenario II, Fig. 21 displays the scatter plot with-seed post-optimisation low noise scenario, and Fig. 22 is a scatter plot with-seed post-optimisation high noise scenario.

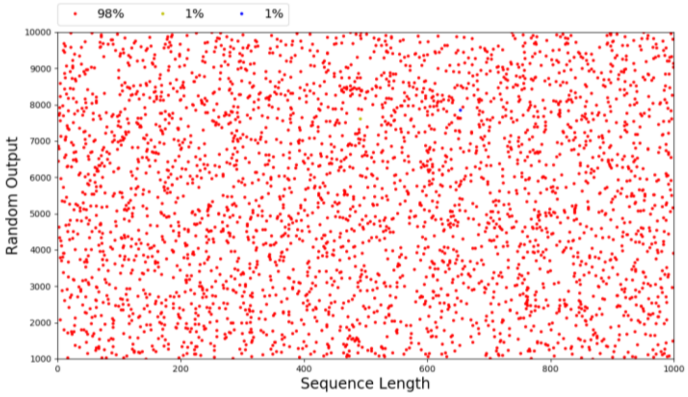


Fig. 12. Random number generation output frequency scatter plot without audio seed

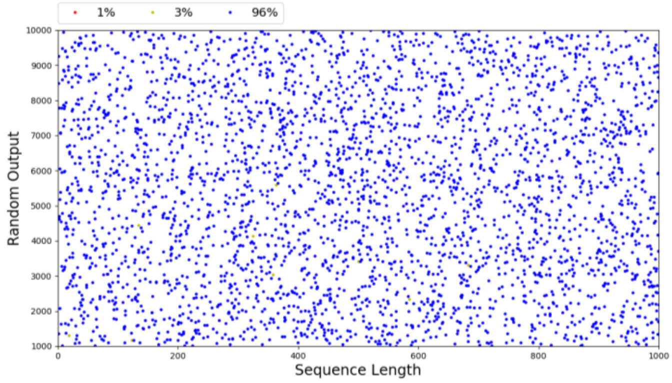


Fig. 13. Random number generation output frequency scatter plot with audio seed, pre-optimisation with noise

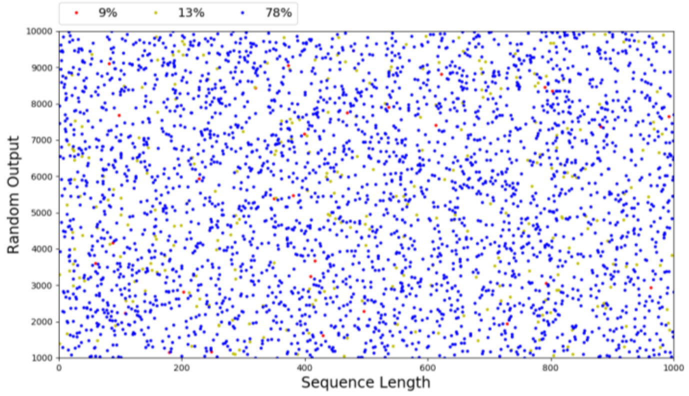


Fig. 14. Random number generation output frequency scatter plot with audio seed, pre-optimisation with no noise

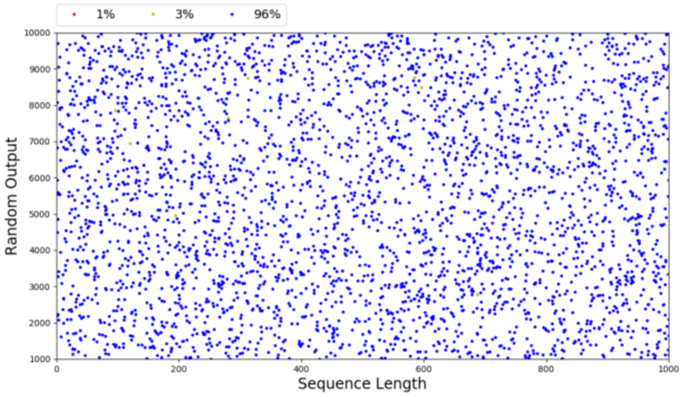


Fig. 15. Random number generation output frequency scatter plot with audio seed, post-optimisation with no noise

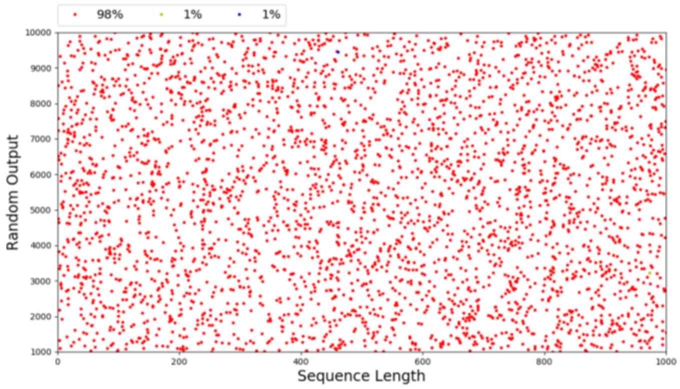


Fig. 16. Random number generation output frequency scatter plot with no-seed pre-optimisation high noise scenario

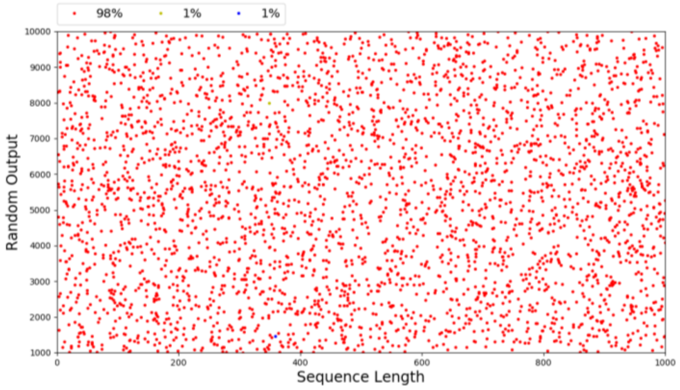


Fig. 17. Random number generation output frequency scatter plot with no-seed pre-optimisation low noise scenario

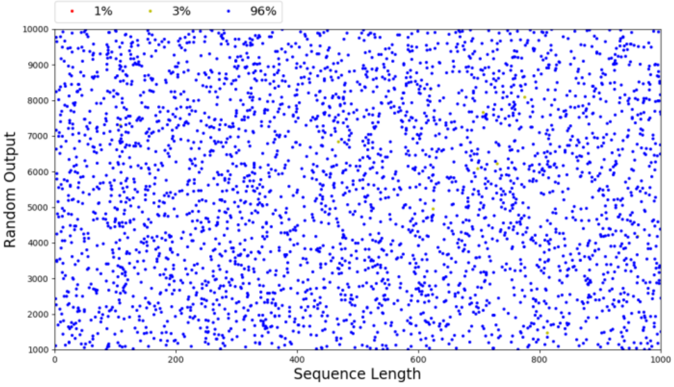


Fig. 18. Random number generation output frequency scatter plot with with-seed pre-optimisation high noise scenario

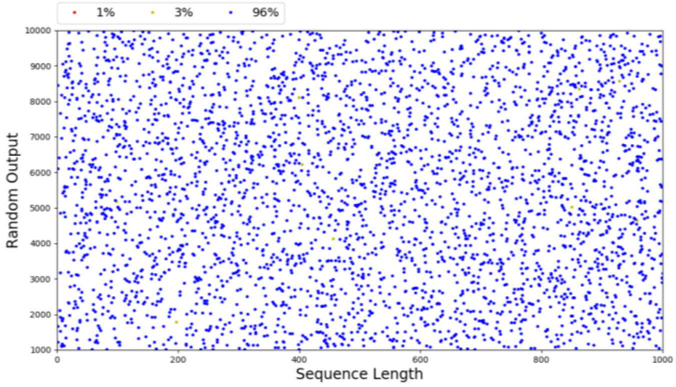


Fig. 19. Random number generation output frequency scatter plot with with-seed pre-optimisation low noise scenario I

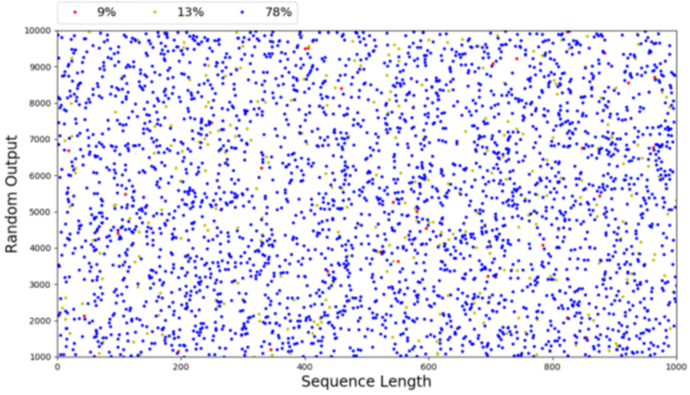


Fig. 20. Random number generation output frequency scatter plot with with-seed pre-optimisation low noise scenario II

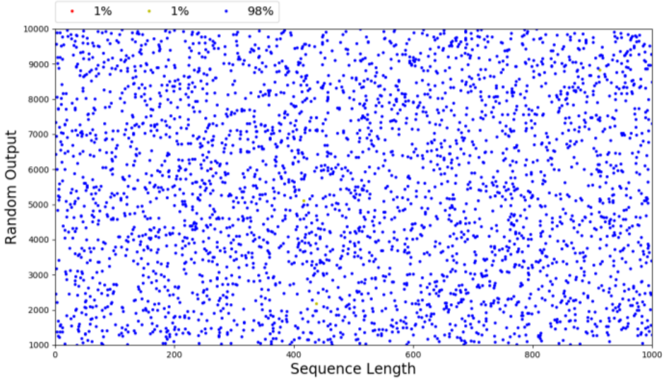


Fig. 21. Random number generation output frequency scatter plot with with-seed post-optimisation low noise scenario

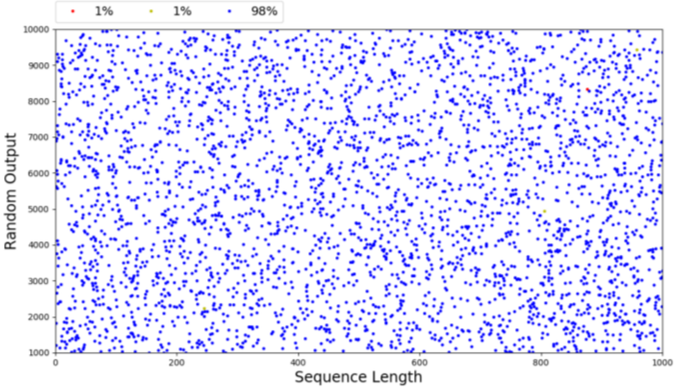


Fig. 22. Random number generation output frequency scatter plot with with-seed post-optimisation high noise scenario

The performances of the three models for the low and high noise scenarios considering no-seed pre-optimisation, with-seed pre-optimisation and with-seed post-optimisation are given in the form of line graphs shown in Figs. 23-28. More specifically, Fig. 23 is the performance of three models low noise scenario I, Fig. 24 shows the performance of three models low noise scenario II, and Fig. 25 gives the performance of the three models low noise scenario III, In a similar fashion, Fig. 26 displays the performance of three models high noise scenario I, Fig. 27 shows the performance of three models high noise scenario II, and Fig. 28 presents the performance of three models high noise scenario III. As observed in the performances of the three models investigated for both low and high noise scenarios – Figs. 23-28, the without seed pre-optimization graphs show minimal variations around the 90-100% recurrence bands. These variations do not appear to be of considerable significance when compared with the variations observed for the with-seed pre-optimisation and with-seed post-optimisation graphs for all scenarios.

The synthesizer considerably drops when there is no presence of ambient turbulence or similar sources of noise as shown by the increase in repetition of randomly generated sequences. This lies the foundation of this work critique, and therefore allows room for improvements by the developed optimisation. This, however, suggests the need for a more rigid definition for the "Low Noise" classification. Based on this experiment, it would be objectively referred to as a "low audio-entropy environment" and not the loosely defined "silent environment." To ensure repeatability and reliability of our results, it was ensured that robust testing was conducted. This involved testing the performances of the OTP generation algorithms, and its three relevant implementations in varied ambient conditions.

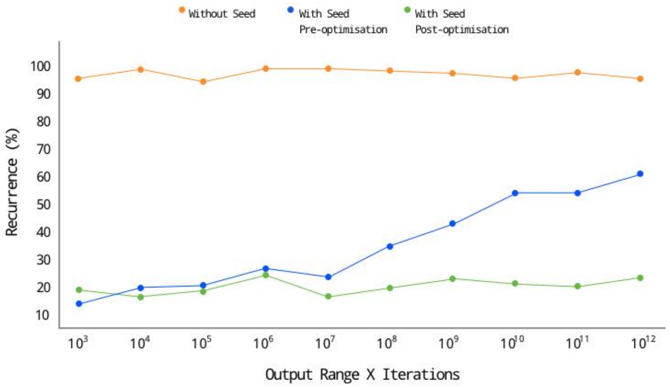


Fig. 23. Performance of three models low noise scenario I

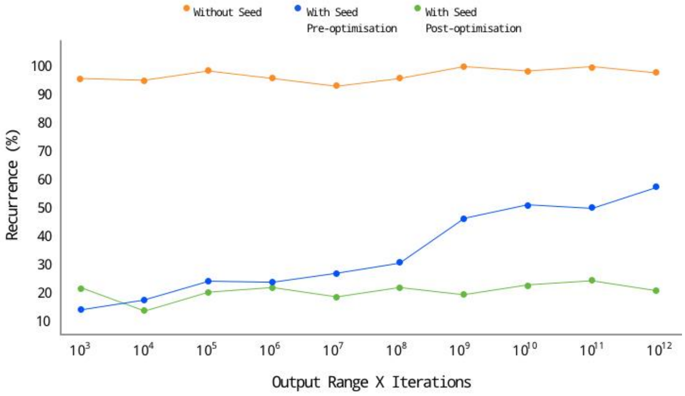


Fig. 24. Performance of three models low noise scenario II

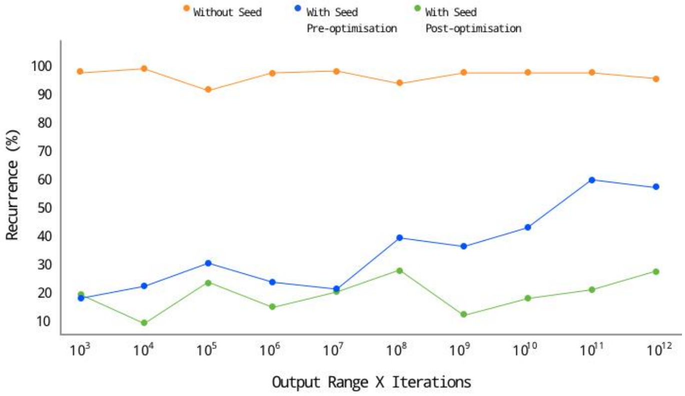


Fig. 25. Performance of three models low noise scenario III

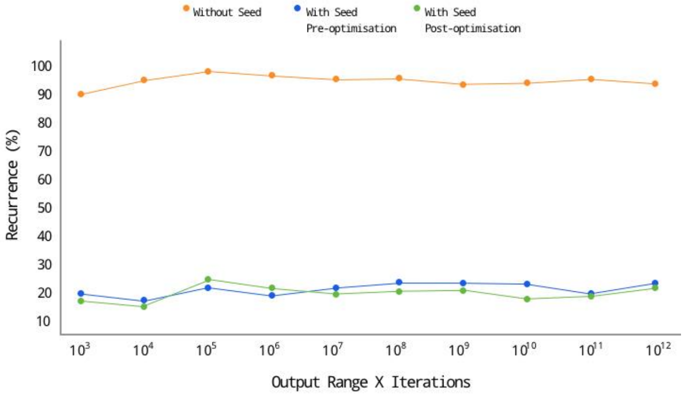


Fig. 26. Performance of three models high noise scenario I

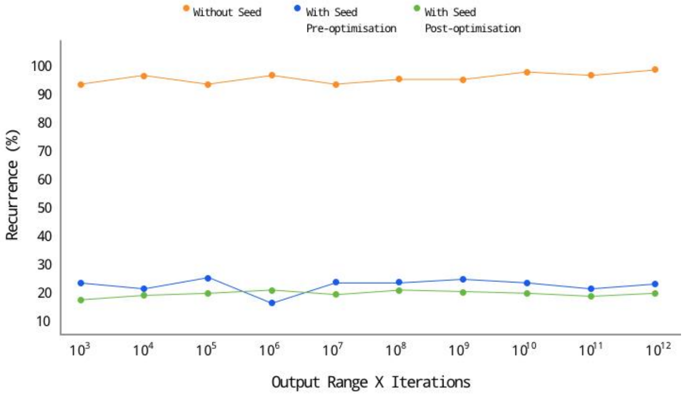


Fig. 27. Performance of three models high noise scenario II

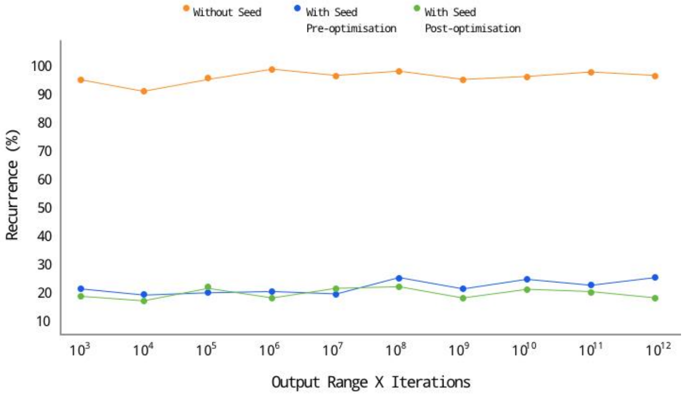


Fig. 28. Performance of three models high noise scenario III

Moreover, the varied scenarios were meant to provide graded level of noise level as classified by atmospheric turbulence. In the first instance, second long audio samples were taken in these varied scenarios and classified into two categories: High Noise and Low Noise. The High Noise samples were taken in environments with noticeable turbulence, such as from moving vehicles by the road side. The Low Noise samples were taken in more serene environments in the software laboratory which could be referred to as a quiet environment. For each scenario, three samples were taken each as shown in Figs. 23-25 for the low noise scenario, and Figs. 26-28 for the high noise scenario. The with-seed pre-optimisation appears to show much more growth in recurrence % when closely compared with the with-seed post-optimisation for the three samples investigated. This implies that the low-noise subtests, the pre-optimisation algorithm fairly performed better than expected, since previous hypotheses led to a critique of the original implementation as being unreliable in low noise levels.

For clarity, a comparative analysis of the tests under low and high noise scenarios are given Table 1. Results show that the performance of the post-optimisation algorithm is poorer than that of the pre-optimisation version. However, after 10^4 iterations, the performance of the two models begin to diverge with the post-optimisation becoming demonstrably superior in performance as the recurrence percentage of the pre-optimisation algorithm continues to increase towards the worst possible value of 99.99% recurrence; a scenario where all random output values are repeated across every iteration.

Table 1. Comparative analysis of tests under low and high noise scenarios.

Iterations	Pre-	Pre-	Post-
	Seed	Optimization	Optimization
10^3	93.11%	14.89%	21.04%
10^4	94.23%	21.47%	14.43%
10^5	94.60%	25.43%	17.21%
10^6	93.72%	31.45%	19.21%
10^7	92.4%	39.6%	18.98%
10^8	93.80%	43.74%	23.01%
10^9	92.90%	49.31%	24.14%
10^{10}	92.33%	61.50%	25.22%
10^{11}	91.10%	65.34	19.21%
10^{12}	90.24%	70.23%	23.19%

Furthermore, a careful analysis of the audio files from the low-noise room revealed that the possible source of disturbance could be some atmospheric turbulence from the ventilation unit in the room, which appears to be silent to the human ear but strong enough to present turbulence and entropy for the algorithm to use even without any optimisation recommendations. Finally, Figs. 24-26 depict the minimal variation in results between 10-20% recurrence when the model is deployed in varying environment, and these show coherence to the predicted model of behaviour. This model predicts that with no entropy addition, without any seed into the Arduino seed function, there will be little or no variance in the poor performance of the recurrence percentages, in the range of 90-100%, which in turns satisfy most of the generated numbers in a given sequence of varying length, and these will be repeated after subsequent iterations of generation.

5. Conclusion and Future Work

In this paper, an ambient noise synthesis using multi-factor protocol has been designed, implemented and evaluated using standard metrics. The performance of the random sequence generator after the introduction of the random seed from ambient noise is quite satisfactorily with values of percentage recurrence between 40-100%. This is relatively stable within the range over many iterations as shown on the logarithm scale of the iterations X output range. However, the results show that in low noise conditions, the pre-optimisation model experiences severe degradations in performance as its values of percentage recurrence approaches the sub-70% region, while the performance of the post-optimisation model remains in the sub-30% region in this condition. This confirms the predictions of underperformance in certain conditions of the pre-optimisation model, and validates the positive contributions of the optimised implementation in comparison with the existing model. Future work would focus on adding multiple factors of authentication such as biometric factoring, and remote password generator server to the current implementation. Finally, by considering best practices for security management systems, a more robust administrative management model could be developed to handle candidate functionalities such as user profile creation, database management of user credentials and log activities of entry requests successes and failures.

Acknowledgements

The authors acknowledge the University of Lagos for supporting this research work. A. L. Imoize was partly supported by a Fulbright Fellowship with the Wireless @ VT Labs, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia VA, USA.

References

[1] R. Bhalla and N. Jeyanthi, *M2U2: Multifactor Mobile Based Unique User Authentication Mechanism*, In A. Abraham et al. (Eds.): Springer Nature Switzerland AG 2020 ISDA 2018, AISC 940, pp. 455–464, 2020.

[2] M. Julian, H. Gareth, and K. D. Mcdonald-maier, “Multi-factor Authentication using Accelerometers for the Internet-of-Things,” *IEEE 2017 Seventh Int. Conf. Emerg. Secur. Technol.*, Canterbury, United Kingdom, pp. 1103–107, 6-8 Sept. 2017.

[3] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla, “Challenges of Multi-Factor Authentication for Securing Advanced IoT (A-IoT) Applications,” *IEEE Netw.*, vol. 33, no. 2, pp. 82–88, 2019.

[4] A. L. Imoize, T. Oyedare, M. E. Otuokere, and S. Shetty, “Software Intrusion Detection Evaluation System: A Cost-Based Evaluation of Intrusion Detection Capability,” *Commun. Netw.*, vol. 10, no. 04, pp. 211–229, 2018.

[5] S. Meena, S. Kapur, V. C. Dhobal, and S. Sethi, *Authentication Scheme Using Sparse Matrix in Cloud Computing*, In A. Abraham et al. (Eds.):

Springer Nature Switzerland AG 2020 ISDA 2018, AISC 940, pp. 43–52, 2020.

[6] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1268–1293, 2015.

[7] I. F. Anyasi and A. L. Imoize, "Information technology and the business communities: A case study of small-scale business enterprises in Nigeria," *Res. J. Appl. Sci. Eng. Technol.*, vol. 2, no. 1, pp. 45–49, 2010.

[8] A. L. Imoize, T. Oyedare, C. G. Ezekafor, and S. Shetty, "Deployment of an Energy Efficient Routing Protocol for Wireless Sensor Networks Operating in a Resource Constrained Environment," *Trans. Networks Commun.*, vol. 7, no. 1, pp. 34–50, 2019.

[9] H. Guo, Q. Zeng, Z. Chen, and M. Zhao, "Bluetooth door lock system based on smart mobile device," *ACM Int. Conf. Proceding Ser.*, vol. 4, pp. 50–62, 2018.

[10] N. H. Ismail, Z. Tukiran, N. N. Shamsuddin, and E. I. S. Saadon, "Android-based home door locks application via Bluetooth for disabled people," *Proc. - 4th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2014*, 28-30 November, Batu Ferringhi, Malaysia, pp. 227–231, 2014.

[11] J. Potts and S. Sukittanon, "Exploiting Bluetooth on Android Mobile Devices for Home Security Application," *In* 2012 Proceedings of IEEE Southeastcon, 15-18 March, Orlando, FL, USA, pp. 30–35, 2012.

[12] L. Kamelia, A. Noorhassan, M. Sanjaya, and W. Mulyana, "Door-automation system using bluetooth-based android for mobile phone Door-Automation System using Bluetooth-Based Android for Mobile Phone," *ARPN J. Eng. Appl. Sci.*, vol. 9, no. 10, pp. 1759–1762, 2014.

[13] M. Kumar, M. Hanumanthappa, S. Kumar, and A. K. Ojha, "Android Based Smart Door Locking System with Multi-User and Multi Level Functionalities," *Int. J. Adv. Res. Comput. Commuunication Eng.*, vol. 5, no. 2, pp. 115–118, 2016.

[14] A. Jain, A. Shukla, and R. Rajan, "Password Protected Home Automation System with Automatic Door Lock," *MIT Int. J. Electr. Instrum. Eng.*, vol. 6, no. 1, pp. 28–31, 2016.

[15] A. Sharma, A. Vijwani, A. Gupta, A. Singhal, and T. Sharma, "Design of a mobile based application for Smart Car Lock System," vol. 5, no. 3, pp. 211–216, 2019.

[16] S. Suryawanshi, B. Patil, V. Mahajan, M. Pawar, and U. Patole, "Smart Door Lock System on Smartphone," *Adv. Comput. Data Process. 2K19)*, vol. 01, no. February, pp. 13–16, 2019.

[17] A. Trisnani, B. F. Barry, H. Santoso, I. M. Putra, and F. A. Saputra, "Smart Door Lock: Anti-Sabotage Door Security System for Restricted Room," *Proc. Sci. Technol.*, vol. 9, pp. 112–119, 2017.

[18] M. R. Navya and P. Ramachandran, "Development of Secured Home Automation using Social Networking Sites," *Indian J. Sci. Technol.*, vol. 8, no. 20, pp. 1–6, 2015.

[19] S. Shin, K. Han, and K. Jin, "Digital Door Lock on the Access Control System using OTP-based User Authentication," *Int. J. Digit. Content Technol. its Appl.*, vol. 7, no. 11, pp. 436–442, 2013.

[20] M. A. Prada-Delgado, A. Vázquez-Reyes, and I. Baturone, "Physical Unclonable Keys for Smart Lock Systems using Bluetooth Low Energy,"*IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, pp. 4808–4813, 23-26 October, 2016.

[21] C. L. Hsu, S. Y. Yang, and W. Bin Wu, "Constructing intelligent home-security system design with combining phone-net and bluetooth mechanism,"*2009 International Conference on Machine Learning and Cybernetics*, Hebei, China, pp. 3316–3323, 12-15 July 2009, 2009.

[22] M. Anitha, A. Babel, A. Kumar, A. Rauniyar, and K. Zahid, *Cloud-based secured QR code for self-service access control system at resort and hotels*, In S.-L. Peng et al. (eds.), Computing and Network Sustainability, Lecture Notes in Networks and Systems 75. pp. 387-395, 2019.

[23] J. Dabhade, A. Javare, T. Ghayal, A. Shelar, and A. Gupta, "Smart Door Lock System: Improving Home Security using Bluetooth Technology," *Int. J. Comput. Appl.*, vol. 160, no. 8, pp. 19–22, 2017.

[24] D. Regis, "Bluetooth Low Energy Door Lock with Ambient Noise Number Generation," *Eng. Div. Grad. Sch. Cornell Univ.*, pp. 1–24, 2016.

[25] T. Liu, H. Lu, and Z. Wei, "Design and Implementation of Intelligent Window Control System Based on Multi-sensor Fusion," in *IEEE 8th Data Driven Control and Learning Systems Conference*, Dali, China, pp. 1368–1372, May 24-27, 2019,

[26] T. Pawlenka and J. Škuta, "Security system based on microcontrollers," *Proc. 2018 19th Int. Carpathian Control Conf. ICCC 2018*, Szilvasvarad, Hungary pp. 344–347, 28-31 May, 2018.

[27] M. Andriansyah, M. Subali, I. Purwanto, S. A. Irianto, and R. A. Pramono, "E-KTP as the basis of home security system using arduino UNO," *Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017*, Kuta Bali, Indonesia, pp. 1–5, 8-10 August, 2017.

[28] L. R. Wilhelmsson, M. M. Lopez, and D. Sundman, "NB-WiFi : IEEE 802 . 11 and Bluetooth Low Energy combined for Efficient

Support of IoT," *2017 IEEE Wirel. Commun. Netw. Conf.*, 2017. San Francisco, CA, USA, 19-22 March, 2017.

[29] S. O. Ajose, R. A. Bakare, and A. L. Imoize, "BER comparison of different modulation schemes over AWGN and Rayleigh fading channels for MIMO-OFDM system," *Int. J. Commun. Networks Distrib. Syst.*, vol. 18, no. 2, pp. 129–147, 2017.

[30] S. O. Ajose, A. L. Imoize, and O. M. Obiukwu, "Bit error rate analysis of different digital modulation schemes in orthogonal frequency division multiplexing systems," *Niger. J. Technol.*, vol. 37, no. 3, pp. 727–734, 2018.

[31] A. Alafia, S. Ajose, and A. Imoize, "A study on low-complexity transmit antenna selection for generalized spatial modulation," *IIUM Eng. J.*, vol. 19, no. 2, pp. 105–117, 2018.

[32] C. J. Hessler, "Method for Mobile Security Via Multi-Factor Context Authentication," *United States Pat.*, no. US 8935769 B2, pp. 1–36, 2015.

[33] P. S. Teh, A. J. Teoh, and S. Yue, "A Survey of Keystroke Dynamics Biometrics," *Hindawi Publ. Corp. - Sci. World J.*, vol. 2013, no. 408280, pp. 1–24, 2013.