Proceedings of the International on Science and Innovative Engineering 2019
24th March 2019, Chennai, India

Consumer Internet of Things Security: Smart Door Lock System

Gowthamy. *J*
Assistant Professor
SRM Institute of Science& Technology
Ramapuram, Chennai
gowthamyjsrm@gmail.com

Kumar Abhishek
Student
SRM Institute of Science& Technology
Ramapuram, Chennai
kabhi481@gmail.com

Devi Subadra. V
Student
SRM Institute of Science& Technology
Ramapuram, Chennai
devisubadra.venkateshan5@mail.com

Rishabh Saxena
Student
SRM Institute of Science& Technology
Ramapuram, Chennai
rishabhsaxena247@gmail.com

***Abstract*— The consumer Internet of Things (IoT) platforms are trending and widely used today. Without the proper implementation of proper security measures, smart home platforms will have problems with their security and privacy. The use case of door lock, connected to the Internet and with ability to enter or exit room based on rights, is used for design and analysis of custom-made systems regarding security. The design, ability and implementation to home automation to unlock a door with authentication through biometric sensors such as fingerprint reader, camera, speech command, or pin via smart phone is presented in the proposed system. The proposed system also provides a dual layered security mechanism against unauthorized entry and hence enhances the integrity of the system. This system thus eradicates the common security threats such as Spoofing, Repudiation and trafficking from evaders especially in the current scenario of industry.**

Keywords- IOT, Security, Smart Home, Speech, Algorithm

I. INTRODUCTION

Smart locks are sure to become every homeowner's favourite. They are very similar to the typical door locks physically and provide more options as alternatives to typical key like keyless locking and entry. Typical locks have only a key to get access but they can easily get lost or left behind but this is not then case with digital door lock as they offer a variety of lock and unlock methods. Generally, codes with the length of 4 to 15 digits are used by the digital lock to provide keyless access to the user. Tokens are also used for authentication when they come into the proximity of smart door lock. They can be easily carried around. Biometrics can also be used to provide seamless experience of smart door lock as they offer enhanced user authentication. Biometric sensors detect user's physical characteristics such as eye scan, voice recognition, fingerprint. These advanced authentication techniques can be used to enhance the security and convenience of door lock system available today.

II. IOT CYBER SECURITY:

IoT systems are part of our daily life cycle and cyber environment. Comparing traditional computers and networks to IoT we can find following differences that influence security:

- IOT devices are usually microcontrollers that has limited capabilities. Because of that state-of-the-art encryption algorithms cannot be implemented in such devices. Only the newest version of microcontrollers has implemented encryption algorithms in hardware. That has negative implications if some defects in algorithms are found then those devices cannot be patched with new versions.
- Manufactures of IoT devices have put the same default credentials in devices during production because it is cheaper to manufacture them. The result was Mirai malware the exploit that to take control of large number of web cameras and similar devices, create IoT-based botnets and run DDoS attacks.
- IoT devices will can be put in the physical space which cannot be secured in every situation like physical attack, it could temper the device. Besides physical alteration of device, attacker could have access to the programming interface. That should also be secured by either providing credentials before programming or by braking fuse that protect writing to memory, so it cannot be programmed again.
- Light weight protocols should be used between IoT devices for inter communication. Concerns about security were very high in earlier protocols. When designing IoT systems security in protocols need to be considered from the beginning of design.
- Some devices are powered by batteries are vulnerable on DoS attacks as intensive communication can drain battery.

ISBN 978-93-81288-18-4

III. RELATED WORK:

Nowadays, there are many commercial smart locks available on the market, based on various technologies for remote communication, typically Bluetooth and Wi-Fi. Some ideas presented in the field are mobile application for (un)locking doors are, a smart door lock system consisting of Raspberry Pi, webcam and speakers, a smart lock system focused on the user recognition to automatically open the door, use Bluetooth technology for the communication between a user and lock. The main difference of our solution, opposed to related solutions, is the use of low-priced ESP modules which, in turn, significantly lowers the price of the proposed locks.

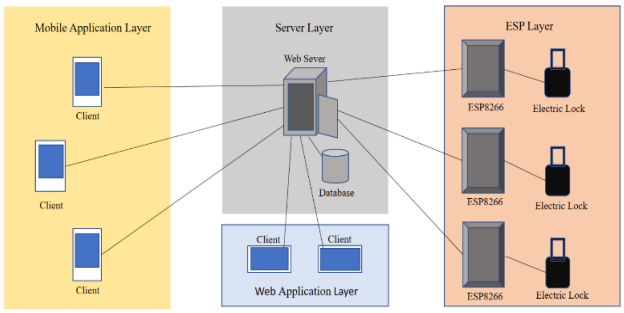
Existing commercial solutions offer cloud-based control of their locks, which is appealing but may also be a significant security issue. In our solution, we offer the possibility to have on-premises control servers inside a local network or a VPN, thus remaining in full control of the multiple lock orchestration. Security is a very important issue in locks of all types and applications, and it is our impression that existing, both commercial and research-driven, solutions do not focus on protection of the locks as they should.

An application can be designed to control home automation. It will not only allow the user to have on off button but also add authentication via speech command or pin. The GUI of application will make it user friendly and give the flexibility to the user. The application will use integrated NFC potential of a smart phone. This acts as a module to open the entry point by a logical link control mechanism. The permission access is granted with respect to password or a code matching.Home security system can also use motion sensors to detect the presence of strangers in the proximity of the building. The sensors used are usually PIR sensors and ultrasonic sensors Users can be notified for any detected movements around the perimeter. RFID is also used in the home security systems. Although its use causes deficiencies in the system which results in user not getting entry or access in the secured building.

IV. System Architecture of Smart Door Lock System:

IoT solutions are fundamentally reshaping the perspective of intelligent workspace which we have known. With the help of interconnected sensors in a restricted area, devices can become more automatic in terms of completing a task and reporting the output back to the user.
Solution's architecture conforms to the most needed requirements regarding horizontal scalability when multiple employees are granted access to multiple areas with different security roles. Architecture is also designed to offer the following features:

- Flexibility – enhancing the user interface and making it simpler.
- Evaluation and prediction – access data will be stored and eventually used for better resource management and further improvement of balance between individual comfort and overall work efficiency.
- Energy savings – optimum use of power supplied.



Smart Lock Architecture

Keeping the above features in mind, the proposed system consists of following parts:

Lock clients (microcontrollers based on ESP8266 hardware platform connected to electronic door lock) with paired electric locks are distributed in the workspace, deployed to listen for commands directed from the central unit. Every lock client only corresponds with one central unit at the time, following the "separation of concerns" principle.

System administrator provides user to access the secured area with the help of security system interface based on smartphones and web applications. The web application which is used for resource management and security administration handling, providing detailed insight into resource usage over the time.
Typical use case for client with access rights to some resource contains actions described below.

- User logs in the system via mobile app. Provided credentials are validated on the server (central unit) and the authentication response is returned to the client.
- Successfully logged in user is offered various resources available to occupy (unlock).
- After the resource is selected by the client, the request is sent to the central unit to be validated. If authorized, user will be given right to use the resource, and the request will be forwarded to the corresponding lock client, eventually unlocking the resource.

Architecture can be described as a union of four layers. Server is the core element of the system, representing the mediator in every communication between end clients and physical locks (many-to-many relationship). It contains business logic and context for the system in whole, manages specified automation rules and provides local data storage for event logs and application states. Server is accessed via web application through which the underlying data and states are managed (security roles handling, resource manipulation, and analysis of the history data).

ESP layer comprises of lock clients based on ESP8266 and electronic lock. They are the essential component of the distributed lock system. They are responsible for handling requests from the server and changing internal states according to received messages through WebSocket to communicate. After the initial handshake TCP connection is kept open, so the data is sent back and forth through the channel with very low latency.

Another advantage of WebSocket technology is underlying TCP protocol with all its implicit benefits,

ISBN 978-93-81288-18-4

ensuring that the packets will be intact upon arrival. It is also important to say how the ESP connects and pairs with the server. When activated, the ESP opens connection to the specified gateway via Wi-Fi. After local IP address is obtained, handshake protocol is initiated to establish WebSocket connection with the server. When the connection is successfully established, the ESP's active state is stored in server's memory to be used for future communication. If the ESP is not active during a long interval of time, server can test a single ESP client by sending a 'ping' request and then listen for response, potentially discovering ESP client's failure.

Mobile application is the visible frontier of the system to end users. Multiplatform implementations (Android OS, iOS) enable access to resources without platform discrimination. Communication between app clients and the server is RESTful, data is wrapped in JSON and transmitted to server in synchronized manner, always expecting response. When the resource is requested, server initiates authorization process based on provided credentials and delegates the request to single lock as previously explained.

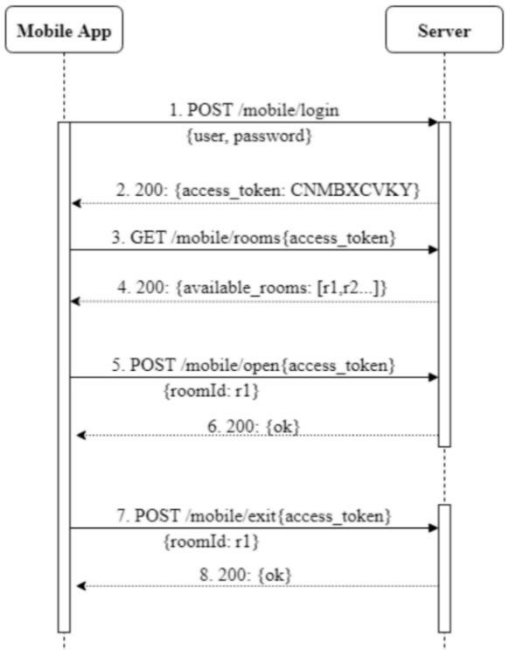
V. COMMUNICATION BETWEEN NETWORK ENTITIES:

A. Client application and server communication

Considering that single mobile application is used infrequently (login, door open), communication with the server is designed on REST principle. User is firstly authenticated with unique credentials (user, password) provided by system administrator. The application sever generates the token which is transferred to the user for authentication purpose. Every succeeding request to server will include client's token in http header (token-based authentication). Except security itself, this approach provides a fine-grained access control to server's resources as the generated token identifies exactly one user. Authenticated user is offered multiple resources from the central unit (defined by assigned roles). User then selects the resource, i.e. the room he wants to open. Central unit receives the request, validates the user's permissions for chosen resource, and eventually delegates the request to ESP in charge for that resource. From a security point of view, basic HTTP tokens are not sufficient for system's trustworthiness – all communication must be secured with SSL (secure sockets layer) or TLS (HTTPS).

B. ESP8266 – server communication

Each lock is paired with a single ESP device used as a gateway for communication with the central unit. After client's request is confirmed as valid, a command message is forwarded to the ESP in charge for the requested resource. This use case includes frequent connections and exchange of messages between server and the ESPs. For that reason, WebSocket technology is used. After a single ESP boots up and becomes active, it must first establish the connection to the local access point via WiFi. Server must be running in the same local network, so it can be addressable from the newly connected ESP. ESP then contacts the server and the SSL handshake is initiated. When the secured channel is ready, the ESP client starts the local WebSocket client and connects to the central unit's WebSocket server. It is worth mentioning that the central unit stores the state of the opened socket connections in internal map-like structure, referenceable by ESP's local IP address. When the end client sends the request for opening a lock, central unit finds a socket connection state, creates the opening command and sends it through this socket channel to the ESP. This way the request is unicasted only to the target ESP, without causing unnecessary network traffic. The assemble of used messages/commands can be extended with various others, e.g. messages for testing the ESP's activity - simply deducing if the ESP client is alive with 'ping' mechanism.



REST communication between mobile client and

VI. SECURITY ANALYSIS AND THREAT MODEL:

The proposed system, like every other locking system, must have high security standards in mind during the design and implementation phases. Security must be observed on all levels and within all elements in the system to provide throughout security. In this section we define possible threats to the proposed system based on an attack tree and show how potential threats are addressed in the system, having in mind the identified attacker goals.

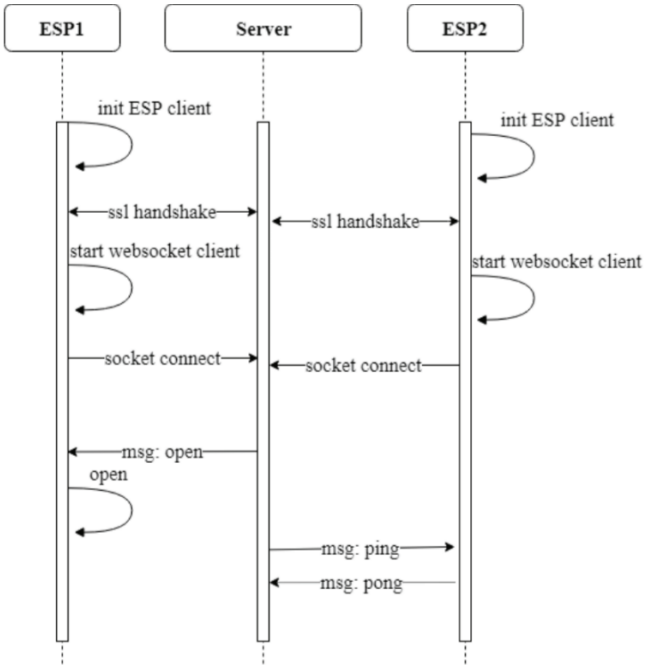
The first and most obvious goal of the attacker is to try and open the lock without privileges. The second goal might be to obtain user privileges by attacking the user device or the authentication server. Finally, the third goal might be to disable the whole system by using one of the attacks that result in Denial of Service (DoS).

A. Opening a lock by gaining privileges

When looking at the system architecture, it is clear that privileges might be obtained from the user device, authentication server or by eavesdropping on the communication between them.

ISBN 978-93-81288-18-4

1. Attacks on user device



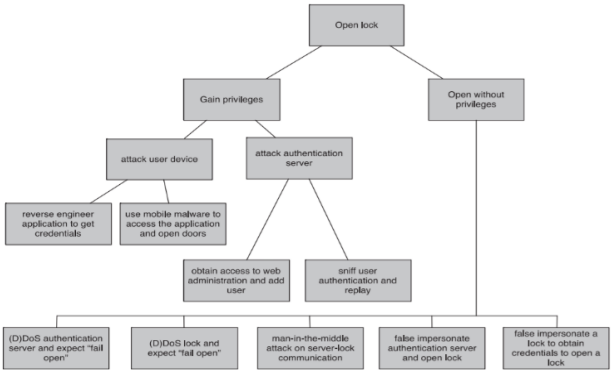
Communication between ESP client and application server

When looking at the user device, the application is used for opening the lock based on user privileges, user being authenticated by a token on the application level. Since the token is stored on the device file system, it might be possible to reverse engineer the application or just browse the file system if the Android device user was given root user privileges (i.e. 'rooted' device). However, since the token is refreshed periodically, this type of attack would be very time limited. Furthermore, reverse engineering an application would provide the attacker with insights into authentication methods on the server, so this should also be mitigated, although this may not result in an actual attack. In order to protect the application from this type of attack, the application should check whether it is running on a rooted device and disable the functionality if so. Regarding reverse engineering, the code should be obfuscated as much as possible, according to best practices, although reverse engineering may be difficult to avoid in the long run.

Another potential attack that focuses on user device is malware that runs on the same device as the legitimate lock control application. The malware might be able to access the application or its files and control it partially, e.g. give commands to open the lock to the application. However, we find this attack very unlikely, other than the fact that the malware might try to steal the token for server access, as suggested earlier.

2. Attacks on the authentication server

Authentication server runs a web application used for registering users and REST services used for communication with the user device and lock. When looking at attacks that try to gain user level privileges in server domain, attackers might try to penetrate the server through web interface intended for human users or by eavesdropping of the communication between user device and authentication server. Penetrating web interface on the authentication server would allow the attackers to change user permissions and add users, resulting in compromising of the complete system. These attacks fall into web application security domain and the proposed web application should be implemented and verified according to OWASP's Application Security Verification Standards. The attacks that rely on eavesdropping are mitigated by the appropriate use of TLS in communication, where all the traffic is encrypted, and the server is authenticated using



Attack tree of the proposed system

trusted certificate. Without the encryption, the attacker might capture the packets containing commands to open specific doors and replay them at any time. The term "any time" is also important; since the application uses time-limited tokens, such attack would be available only for a limited time interval.

B. Opening a lock without privileges

Another main group of attacks is focused on methods that try to open or disable the lock without gaining the actual specific user privileges. These attacks fall in the category of network-based attacks and false impersonation.

1. Using attacks to achieve denial-of-service

Denial of service (DoS) typically results in the complete unavailability of the targeted system for a period of time. However, locks on physical doors are especially prone to such attacks due to the safety concerns. While most information systems tend to "fail safely" which means that when they fail, the data and services remain safe, systems that control doors should "fail open". This is important in the case of fire, for example, where the doors should be always able to open even when the lock control system is not in function. So, DoS attacks on authentication server would result in unavailability of the server to issue commands to open the doors. In that case, the doors should be able to open physically, which is indeed the case in the proposed system where all doors can be also open by using a standard key from outside or just by a knob from inside of the room. In this sense, we cannot claim that we can mitigate DoS attack since they are practically always possible, especially if they are distributed. But, even in that case, the doors could be open in a traditional manner.

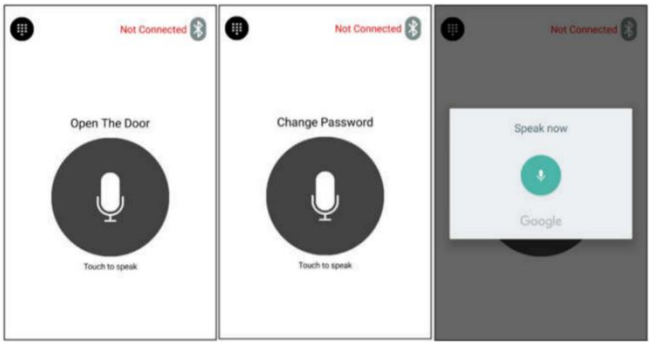
When looking at the other side of the system, the lock, a DoS attack would be unsuccessful since the locks do not have open connection listeners, they are all in the private

ISBN 978-93-81288-18-4

network and are, as such, invisible to the attackers from the outside and they only communicate to the authentication server as described in previous sections.

2. False impersonation attacks

False impersonation attacks may be the weakest point of the proposed system. When looking at the architecture,



User Interface for Speech Command

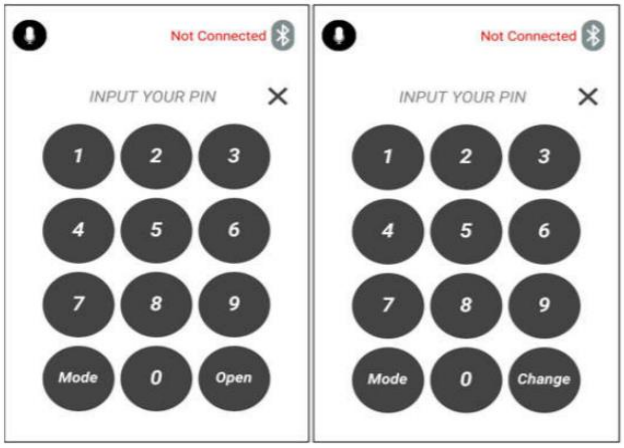
the attackers may falsely impersonate as a lock or as an authentication server.

False impersonating as a lock to the authentication server might be possible if the attackers had access to credentials stored within the lock. This would require reverse engineering of the ESP8266 module and application on the lock, which would be possible if the attacker had physical access to the device. Even then, the attacker would only achieve that the real lock would be unable to connect to the authentication server (similar to partial DoS) which would turn it into an "ordinary" lock. However, we find this attack highly unlikely due to the need of physical access, easy discovery and recovery.

On the other hand, false impersonation of the authentication server is a real threat to the proposed system as currently implemented. As explained previously, the locks connect to the server and authenticate with their specific credentials. The server can be authenticated using a certificate, but the modules used for controlling the lock are ESP8266 which are unable to process cryptography functions and freezes c the process of validation of the server. Therefore, if the attacker managed to false impersonate an authentication server by one of the techniques from the domain of DNS spoofing this attack might be possible. In further implementations the plan is to use new modules, ESP32, that offer HSM modules with the encryption ability.

3. Man-in-the-middle attacks

It is one of the types of widely used attacks on the security system. It implies that the attacker managed to put a malicious node between authentication server and a lock, using one of the several methods for MITM. This would allow the attacker to eavesdrop and change the packets that are transferred. The attacker can dismantle the sequence of packets transferred. The attacker might try to do a simple replay of the command to open the lock at any time or change the packets to obtain some other insights into the system and lock behavior. This type of attack is a problem, the same as false impersonation of the server, since the currently used ESP8266 module cannot check the



User Interface for PIN Authentication

authenticity of the server. Only possible solution at this point is to use the ESP32 with cryptographic functions.

VII. IMPLEMENTATION:

This research work makes a system for unlocking the door by android phone. Two methods are tested for door automation system, speech command authentication and pin authentication. Users can choose one of them from the android application which is preferred to be used for control to unlock the door.

A. Speech Command

1. Android Application

This research work tested the automation via voice. Speech command is used for control to unlock the door. The android application of this work is created with MIT App Inventor web based. It already has google speech to text library, called speech recognizer. This program is using the speech to text of google library. The spoken word as a command to unlock the door is converted to text uses google speech function in MIT App Inventor. The serial connection between android application and Arduino microcontroller is also using Bluetooth. The both Bluetooth address must be paired first until the interface of android application turn become connected from not connected. Then the text from the speech command will be sent to the Arduino microcontroller via Bluetooth and will be authenticated if it is the correct command or not.

2. Hardware Implementation

The main part of the door automation hardware is the Arduino microcontroller. It receives the data from the android application and control the supporting components. Transfer of speech data between android application and Arduino microcontroller takes place with the help of Bluetooth. Arduino microcontroller has non-volatile memory storage, EEPROM. This memory is used to save the command password in order to the stored memory will not lose even when there is no power supply give in.

B. PIN Authentication

1. Android Application

ISBN 978-93-81288-18-4

Another work besides speech authentication, the application gives another choice to unlock the door by pin authentication. Same as in the speech command interface, the GUI of the android application is made first. When a new project started, the view of MIT app inventor web based will be in the designer tab. Layout and user interface can be drag from palette column and drop to the screen viewer in the middle. The properties also can be set through the properties column. We can remap the position of all components including user interface and layout. In this door automation project, the android application has connectivity of Bluetooth client in order that the smartphone can search the nearest other active Bluetooth in the range. The purpose is connecting the android smartphone with the Bluetooth module in the door automation hardware. When it is connected, the door can control from the application via android smartphone.

When the android smartphone Bluetooth is turn on, the Bluetooth client function of the program start to make list all paired Bluetooth and save the address and name. From that paired Bluetooth list, the address and the name of Bluetooth hardware is selected in order to make connection between the both. There will appear warning text if the connection failed.

Speech Command Algorithm
1. Initialize the password of speech command
2. Call speech to text google library
3. Send the text from spoken word via Bluetooth
4. Receive the text data in the Arduino microcontroller
5. Check the protocol, if the first input character is #, the data is true for the system
6. If not, do nothing
7. If true continue check, the next flag must be 0 to indicate the use of speech command authentication
8. Then continue check, if the next flag serial setting is 1, it is the command to set new command password, then update the received detected text in the next serial data as saved password in the EEPROM
9. If the next flag serial setting is 0, it is the command protocol to open the door.
10. Do the authentication, If the command password received are exactly the same with the saved password, the relay will turn HIGH to control the solenoid to open the door
11. If not, shows the warning in the android application

Speech Command Algorithm

2. Hardware Implementation

The pin data, which is sent by the android application via smartphone Bluetooth, will be received by the Arduino microcontroller via Bluetooth module which has been installed on it. Arduino microcontroller has non-volatile memory storage, EEPROM. This memory is used to save the pin password in order to the stored memory will not be lost even when there is no power supply give in. The protocols are defined correlated with the android application. The corresponding received data is identified by the header data, using #. Data with different protocol will not be processed. Data with the # header will continue to the next authentication. The next flag is used to distinguish the speech data or pin data. As explained before, speech authentication uses 0 as flag, while pin authentication uses 1 as flag. The password setting is sent through protocol 1 after #1. While the open the door function is sent with 0 after #1. So, the protocol of pin authentication will be like #1 then 0/1 followed with the pin.

VIII. CONCLUSION:

PIN Authentication Algorithm
1. Initialize 4 digits pin code in the Android Application
2. Send the pin data via Bluetooth
3. Receive the pin data in the Arduino microcontroller
4. Check the protocol, if the first input character is #, the data is true for the system
5. If not, do nothing
6. If true continue check, the next flag must be 1 to indicate the use of pin authentication
7. Then continue check, if the next flag serial setting is 1, it is the command to set/update the pin code, then update the received 4 digits pin code in the next serial data as a saved pin in the EEPROM
8. If the next flag serial setting is 0, it is the command protocol to open the door.
9. Do the authentication, If the 4 received digit pin are exactly same with the 4 digits pin saved, the relay turn HIGH to control the solenoid to open the door
10. If not, shows the warning in the android application

PIN Authentication Algorithm

In this paper we have explained about security problems in smart door lock systems. Those problems will be unmanageable with the increasing number of connected devices if they are not addressed early. During design of IoT system security considerations needs to be included from the start. The second thing is that IoT devices are usually constrained devices which does not have capabilities as normal computers. The newest devices have hardware implementation of cryptographic algorithms and secure communication protocols. As for the use case analysed in the paper, we have found some vulnerabilities that will be improved in future work. A user-friendly technology is deployed to render the entirety door automation system with its process to be controlled. Speech command authentication and pin authentication is used to test the automation work. In speech command authentication, google speech library helps the spoken words to be converted to text. By this speech command interface can simplify the use of the application and also speech command can add security function, so does with the pin. The password can be changed by the user as desired. To ensure communication between smart devices are corresponding, the data communication protocol is established.

IX. REFERENCES:

I. Yan Meng, Wei Zhang, Haojin Zhu, and Xuemin (Sherman) Shen, "Securing Consumer IoT in the Smart Home: Architecture, Challenges, and Countermeasures", IEEE Wireless Communications, December 2018a

II. Wazir Zada Khan, Mohammed Y Aalsalem, and Muhammad Khurram Khan, "Communal Acts of IoT Consumers: A Potential Threat to Security & Privacy", 0098-3063 (c) 2018 IEEE

III. Marko Pavelić, Zvonimir Lončarić , Marin Vuković, Mario Kušek, "Internet of Things Cyber Security: Smart Door Lock System", 978-1-5386-7189-4/18/$31.00 ©2018 IEEE

ISBN 978-93-81288-18-4

IV. Lia Kamelia, Mufif Ridlo Effendi, Delingga Ferial Pratama, "Integrated Smart House Security System Using Sensors and RFID",978-1-5386-6163-5/18/$31.00©2018 IEEE

V. Retha Dinar Hayu Arifin, Riyanarto Sarno, "Door Automation System Base on Speech Command and PIN using Android Smartphone", 978-1-5386-0954-5/18/$31.00©2018 IEEE

VI. R. Jenifer Prarthana, A.Mohamed Dhanzil,N.I0. Mahesh , S.Raghul, "An Automated Garage Door and Security Management System", 978-1-5386-0965-1/18/$31.00 ©2018 IEEE

VII. Sumantec, Executive Summary (2018 Internet Security Threat Report), ISTR volume 23, March 2018

VIII. John R. Delaney, "The Best Smart Locks of 2018", PCMag India, Nov. 6, 2018

IX. Shuaike Dong, Menghao Li ,Wenrui Diao, Xiangyu Liu, Jian Liu, Zhou Li,Fenghao Xu, Kai Chen , Xiao Feng Wang and Kehuan Zhang, "Understanding Android Obfuscation Techniques: A Large-Scale Investigation in the Wild", arXiv:1801.01633v1 [cs.CR] 5 Jan 2018.

ISBN 978-93-81288-18-4

JAWAHAR ENGINEERING COLLEGE

Organization of Science and Innovative Engineering & Technology
ISBN 978-93-81288-18-4