Secure wireless controller for hand-held remote operation of Traffic signals in peak hours.

SMART INDIA HACKATHON 2022

By: ***Team Super 6*** *– KLS Gogte Institute of Technology, Belgaum.*

Introduction:

Traffic in India is a changing scenario. It varies from state to state. Compared to other countries India has a very versatile traffic control system. The road rules are different when compared to the other roads.

In India, most of the roads are still under construction and the roadways are in the front of the development.

This gives us the chance of developing the traffic system and make proper roadway transportation in the future.

During peak hours, these roads get filled with people, all travelling to reach their destinations quickly. But there is a possibility that during such times, the irregular traffic volume can be a reason for accidents. Managing traffic during such time is a crucial step that needs to be taken to avoid any kind of accident.

Let us consider another scenario. A VIP is supposed to travel from a place to an important meeting. Supposedly this happens during rush hours, there need to be special steps taken to make sure the travel is safe and secure.

So, there is a need to have an environment that facilitates all the required features for efficient, secure and smart traffic control. That is why we built our system.

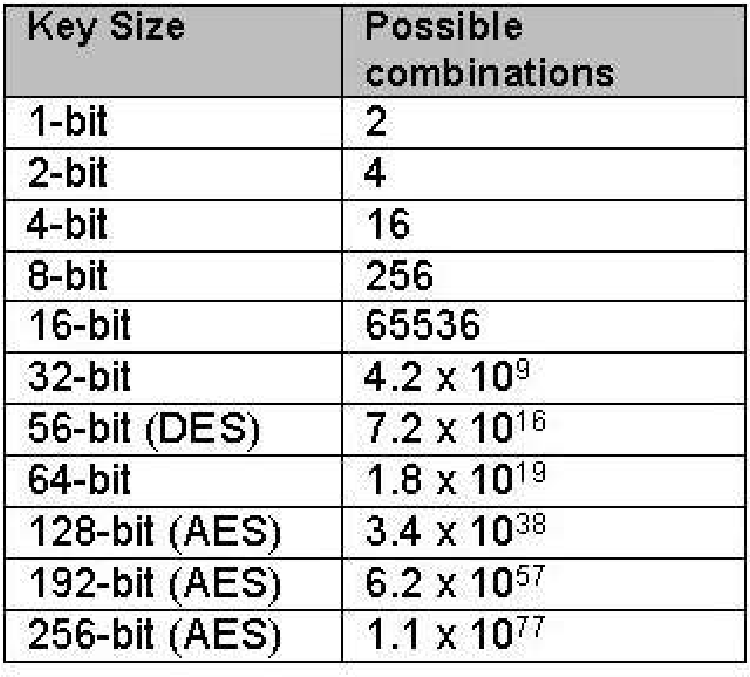
Our device is an '**All in One**' solution for traffic management systems. It is built on top of the ZigBee architecture, which enables it to create its own ecosystem. The system in itself forms an IoT network, which is smart and reliable.

The Security:

The networking devices form a secure and encrypted network so that any form of sniffing is avoided. This system uses the Advanced Encryption Standard (AES) which specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. It is one of the most frequently used encryption systems that is known for its computationally secure algorithm. To date, there have been no reports of it being cracked.

The reason for this statement can be proved with the following argument:

The following table shows the key size and their possible combinations:



Key Size	Possible combinations
1-bit	2
2-bit	4
4-bit	16
8-bit	256
16-bit	65536
32-bit	4.2×10^{9}
56-bit (DES)	7.2×10^{16}
64-bit	1.8×10^{19}
128-bit (AES)	3.4×10^{38}
192-bit (AES)	6.2×10^{57}
256-bit (AES)	1.1×10^{77}

Looking at the table, we can see that our system, which implements a 128-bit key has around 3.4×10^{38} possible key combinations.

Let us try to *analyse* the time required to crack this key using the Brute Force method:

Consider a Fast supercomputer:

10.51 Pentaflops = 10.51 x 1015 Flops [Flops = Floating point operations per second]

No. of Flops required per combination check: 1000 (assuming)

No. of combination checks per second

= (10.51 x 1015) / 1000 = 10.51 x 1012

No. of seconds in one Year

= 365 x 24 x 60 x 60 = 31536000

No. of Years to crack AES with 128-bit Key

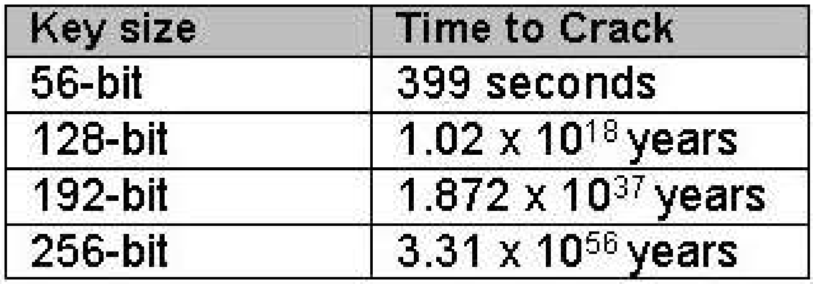
= (3.4 x 1038) / [(10.51 x 1012) x 31536000]

= (0.323 x 1026)/31536000

= 1.02 x 1018

~= 1 billion billion years

Using these calculations, we can observe the time required to crack the key as:



Key size	Time to Crack
56-bit	399 seconds
128-bit	1.02 x 10^{18} years
192-bit	1.872 x 10^{37} years
256-bit	3.31 x 10^{56} years

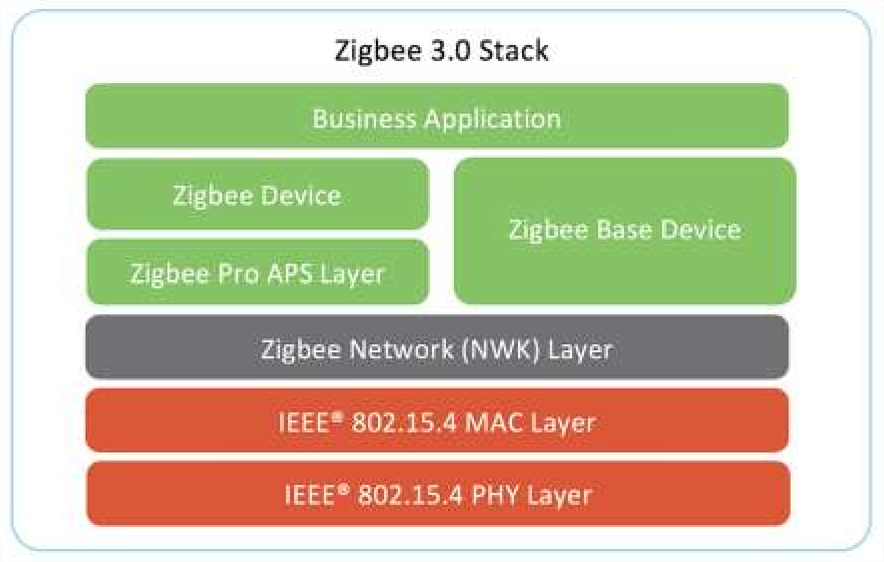
So, by the above calculations, we can say that the AES Encryption is a computationally safe algorithm, in the sense that the energy and resources spent on cracking the key are more.

The ZigBee:

zigbee

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless IoT networks. The Zigbee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz.

The ZigBee architecture is as follows:



The latest Zigbee 3.0 protocol is designed to communicate data through noisy RF environments that are common in commercial and industrial applications.

Version 3.0 builds on the existing Zigbee standard but unifies the market-specific application profiles to allow all devices to be wirelessly connected in the same network, irrespective of their market designation and function.

Connecting Zigbee 3.0 networks to the IP domain opens up monitoring and control from devices such as smartphones and tablets on a LAN or WAN, including the Internet, and brings the true Internet of Things to fruition.

Zigbee protocol features include:

- Support for multiple network topologies such as point-to-point, point-to-multipoint and mesh networks
- Low duty cycle – provides long battery life
- Low latency
- Direct Sequence Spread Spectrum (DSSS)
- Up to 65,000 nodes per network
- 128-bit AES encryption for secure data connections
- Collision avoidance, retries and acknowledgements

Zigbee enables broad-based deployment of wireless networks with low-cost, low-power solutions. It provides the ability to run for years on inexpensive batteries for a host of monitoring and control applications.

The reason for using ZigBee is as follows:

- It provides built-in support for AES Encryption.
- Low Power Technology.
- High-density connectivity options.
- Automatic data handling support, including retires, acknowledgements etc.

The ATmega 328 AVR Boards:

The ATmega328 is an Atmel 8-bit AVR RISC based microcontroller that combines a 32KB Flash memory, 1KB EEPROM, 2KB SRAM, 23 IO lines and many other features.

It works in the voltage range of 1.8 volts and 5.5 volts. It has a maximum clock speed of 20 MHz, giving a maximum throughput approaching 20 MIPS. This means that the processor can process around 1 instruction per clock cycle, giving it a good balance between power consumption and processing speed.

Due to its design being based on the RISC architecture, it is preferred for use in simple to medium complex designs. It provides an 8-Bit bus for data and has features such as support for UART that can be used to communicate with the ZigBee.

The microcontroller can be programmed in Embedded C. The tools used for programming include a programmer and an IDE for writing and compiling the program. The IDE includes all the necessary tools for writing, modifying, developing and deploying programs in AVR devices.

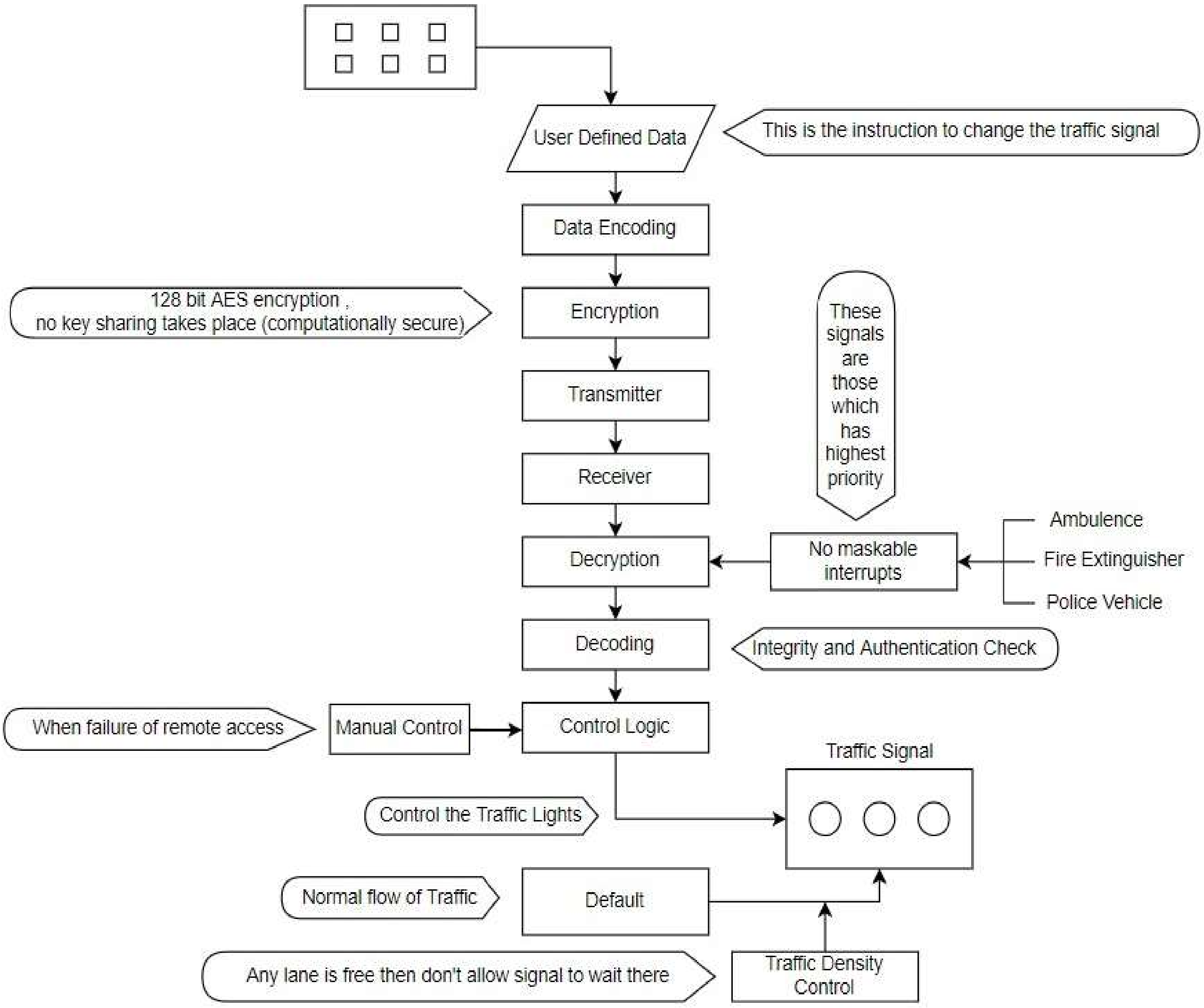
This IDE is the Microchip Studio, a free tool provided by the manufacturer of ATmega chips.

Among the many applications of this chip, one of the notable uses is found in Arduino. Although it contains a bootloader, there are free many free and open-source libraries for the board, which can be ported to work directly on the chip itself.

Having such a big collection of libraries gives an added advantage in faster development time. The chips are budget-friendly too.

Working of the System:

The operation of the system can be understood by the following flowchart:


User Defined Data
This is the instruction to change the traffic signal
Data Encoding
128 bit AES encryption ,
no key sharing takes place (computationally secure)
Encryption
Transmitter
Receiver
These signals are those which has highest priority
Decryption
No maskable interrupts
Ambulence
Fire Extinguisher
Police Vehicle
Decoding
Integrity and Authentication Check
When failure of remote access
Manual Control
Control Logic
Traffic Signal
Control the Traffic Lights
Normal flow of Traffic
Default
Any lane is free then don't allow signal to wait there
Traffic Density Control

From the above diagram, we can see an overview of the working of the system.

In brief, the system works as follows:

1. There is a handheld device with the policeman.
2. This handheld device is firstly approved by the control system for security purposes.
3. Once permitted, the remote is able to control the traffic junction.
4. To change the traffic signal, the policeman feeds the input through that handheld device.
5. The remote(handheld device) uses an encryption algorithm to encode the command signal.
6. This signal is sent to an XBee module.
7. This XBee module is connected through the Controller network. This then encrypts the signal a second time and transmits it into the network.
8. The end traffic signal receives this message, decodes it and checks for its validity.
9. Once validated, the end controller decodes the encrypted command signal and processes it. This sets the traffic lights as per the instruction.

Features:

The system has mainly the following notable features:

- Supports AES 128bit encryption.
- Supports Data validation and authentication.
- The system is portable and less complex due to wireless connectivity.
- Due to the simple design philosophy, it is easy to use.
- Low power consumption.
- It can control multiple traffic lanes.

- Ease of adding or removing any new junctions in the system.
- Low development and implementation time and cost.

Applications:

- Smart traffic light control systems can be implemented using this technique.
- In case of an emergency, such a device can help to override traffic signals to divert the traffic elsewhere.
- In Peak Hours, wrong Traffic diversion can cause accidents, so to avoid such accidents this system can be implemented.
- Using this system, traffic can be controlled to provide a faster way for emergency vehicles (ambulance, fire brigade, etc) to navigate.
- Can be used to develop an autonomous traffic control system

Team Members:

1	Darshan Shivsingh Patel	2gi20ec407@students.git.edu
2	Akshaykumar B Hiremath	2gi20ec402@students.git.edu
3	Chelsi. S. Jain	2gi20ec406@students.git.edu
4	Komal D. Latukar	2gi20ec412@students.git.edu
5	Shivani S. Pujeri	2gi20ec422@students.git.edu
6	Snehal. S. Kanbargi	2gi20ec424@students.git.edu

Mentors:

1	Praveen U. Kalkundri	pukalkundri@git.edu
2	Dr. Veena Desai	veenadesai@git.edu