

A Novel Methodology for Authentication and Car Security Using Image Processing

Vasanth Kumar N, Pradeepraj G, Praveen N, Venkat S, Jeya anusuya S

Department of Electronics and Communication Engineering, T.J.S Engineering College, Peruvoyal, India

Department of Electronics and Communication Engineering, T.J.S Engineering College, Peruvoyal, India

Department of Electronics and Communication Engineering, T.J.S Engineering College, Peruvoyal, India

Assistant Professor, Department of Electronics and Communication Engineering, T.J.S Engineering College,
Peruvoyal, India

Associate Professor, Department of Electronics and Communication Engineering, T.J.S Engineering College,
Peruvoyal, India

ABSTRACT: In the modern world the roadway communication has become the major part in the transportation. In these days usage of car transportation has become high and also the car theft and the drivers without license has become the major concern in our country. In the statistics 2017, motor vehicle theft rate for India was 13 cases per 100,000 population. Motor vehicle theft rate of India increased from 7.2 cases per 100,000 population in 2004 to 17 cases per 100,000 population in 2017 growing at an average annual rate of 6.95 %. And in addition to that rate of drivers driving the car without license results in the deaths of the innocent people. Out of the 5 lakh odd accidents across the country, close to 4 lakh of them involved people with a regular license while the rest involved those with a learner's license or without a license. In other words, 1 out of every 5 accidents involved those without a regular license. Out of the accidents that involved persons without a license, 25% occurred in Madhya Pradesh and 17% in Uttar Pradesh. So, the government has no advanced technique to stop the car theft and the drivers without the license which was going to overcome by our proposed project.

KEYWORDS: RFID Tags, RFID Readers, Sensors, Camera, Microcontroller ,Image Processing

I.INTRODUCTION

In the proposed system we are overcoming the major drawback by using fully automated mechanism. In our project we are using the IR sensor which will detect the whether the person is sitting on the driver seat and the details will be sent the microcontroller and if the IR sensor detects the person then he has insert the key the user has to keep the RFID tag in the RFID reader and in that the driver's license details is fed in that and it reads the details of the driver and send the details to the microcontroller through the ZIGBEE module. From the microcontroller the command will be send to the camera and it will capture the iris of the driver and then the captured iris will be compared with the iris that fed in the RFID tag. If both the details are matched then the car mechanism will turn ON the car and if the details doesn't match then there will be a buzzer which will give the alarm. By this setup we can check the driver license. Addition to this setup we are adding checking mechanism in which there will be a checking lane in the road which will check the crossing car detail and also the detail of the driver through the ZIGBEE module and it will be uploaded in the server through the IOT module. And if the IR sensor didn't sense the presence of person then the details will be sent to the microcontroller and from there car will be immediately turned off by the help of motor mechanism. From that we are able to catch the car theft and the driver without license which will reduce the rate of accidents.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Special Issue 2, March 2019

II.LITERATURE SURVEY

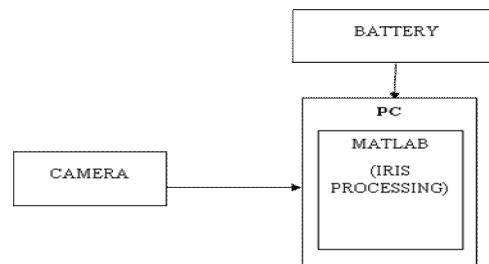
The design of unified, efficient and lightweight cryptographic platform for resource-constrained on-board devices such as sensors, microcontrollers and actuators in the context of Internet of Vehicles (IoV) remains an open and challenging problem, for both academic and industry. Elliptic Curve Cryptography (ECC) is considered as a promising encryption algorithm for the next generation communications, as it could provide the same strong security level using relatively smaller key size when compared to the currently used RSA algorithm.

However, traditional ECCs have the disadvantage of using a fixed curve, making it very easy to be intensively analyzed while being hard to construct a united platform for on-board devices with processors of different instruction lengths. To mitigate the above problem, this paper suggests a dynamic scalable elliptic curve cryptosystem. To synchronize the curve in use, a curve list of different security levels is generated and preserved on both parties. Since both parties randomly choose the curve and the prime number, a extra security level could be provided, so that the security level can still remain the same even using smaller key sizes, while the computation efficiency will be enhanced and the power consumption will be reduced, which is especially suitable for the application in on-board embedded devices. Detailed experimental results illustrate that the presented scheme improves the efficiency by 30% in average when compared with traditional ECC implementations on a similar security level. Therefore, the proposed scalable ECC scheme as a unified cryptographic platform is more economic for these on-board devices in vehicle

III.BLOCK DIAGRAM

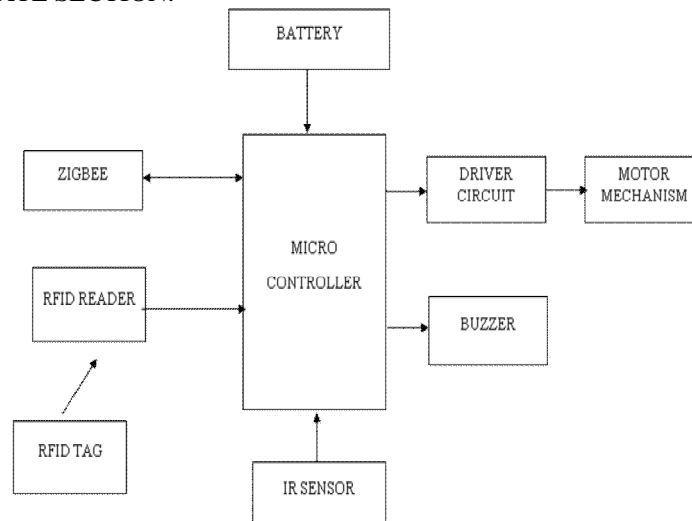
CAR SECTION:

1) IMAGE PROCESSING PART:



2)VEHICLE CONTROL PART

CHECK POST / TOLL GATE SECTION:



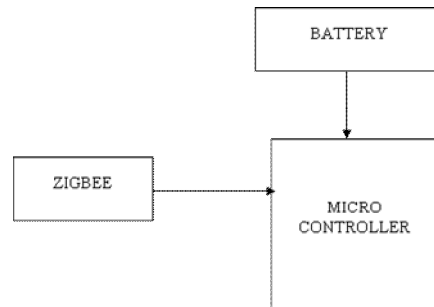
International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Special Issue 2, March 2019

CHECK POST / TOLL GATE SECTION:



IV. DESCRIPTION OF HARDWARE AND SOFTWARE

1. HARDWARE DESCRIPTION

1.1 Power Supply:

a) Transformer:

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op-amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

b) Bridge Rectifier:

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

The path for current flow is from point B through D1, up through RL, through D3, through the secondary of the transformer back to point B. this path is indicated by the solid arrows. Waveforms (1) and (2) can be observed across D1 and D3.

One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through RL, through D2, through the secondary of T1, and back to point A. This path is indicated by the broken arrows. Waveforms (3) and (4) can be observed across D2 and D4. The current flow through RL is always in the same direction. In flowing through RL this current develops a voltage corresponding to that shown waveform (5). Since current flows through the load (RL) during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

1.2 Microcontroller

All the functions required on a single chip. A microcontroller differs from a microprocessor, which is a general-purpose chip that is used to create a multi-function computer or device and requires multiple chips to handle various tasks. A microcontroller is meant to be more self-contained and independent, and functions as a tiny, dedicated computer.

They are typically designed using CMOS (complementary metal oxide semiconductor) technology, an efficient fabrication technique that uses less power and is more immune to power spikes than other techniques. There are also

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Special Issue 2, March 2019

multiple architectures A microcontroller is an integrated chip that is often part of an embedded system. The microcontroller includes a CPU, RAM, ROM, I/O ports, and timers like a standard computer, but because they are designed to execute only a single specific task to control a single system, they are much smaller and simplified so that they can include used, but the predominant architecture is CISC (Complex Instruction Set Computer), which allows the microcontroller to contain multiple control instructions that can be executed with a single macro instruction. Some use a RISC (Reduced Instruction Set Computer) architecture, which implements fewer instructions, but delivers greater simplicity and lower power consumption. Early controllers were typically built from logic components and were usually quite large. Later, microprocessors were used, and controllers were able to fit onto a circuit board. Microcontrollers now place all of the needed components onto a single chip. Because they control a single function, some complex devices contain multiple microprocessors.

1.2.1 PIC16F877A:

The term PIC, or Peripheral Interface Controller, is the name given by Microchip Technologies to its single – chip microcontrollers. PIC micros have grown to become the most widely used microcontrollers in the 8- bit microcontroller segment.

The PIC16F877A CMOS FLASH-based 8-bit microcontroller is upward compatible with the PIC16C5x, PIC12Cxxx and PIC16C7x devices. It features 200 ns instruction execution, 256 bytes of EEPROM data memory, selfprogramming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, a synchronous serial port that can be configured as either 3-wire SPI or 2-wire I2C bus, a USART, and a Parallel Slave Port.

2. SOFTWARE DESCRIPTION

2.1 MATLAB

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos. At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

- Introduction - describes the components of the MATLAB system.
- Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.
- Manipulating Matrices - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.
- Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.
- Programming with MATLAB - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

2.2 EMBEDDED C PORTABILITY

By design, a number of properties in Embedded C are left implementation defined. This implies that the portability of Embedded C programs is not always guaranteed. Embedded C provides access to the performance features of DSPs. As not all processors are equal, not all Embedded C implementations can be equal. For example, suppose an application requires 24-bit fixed-point arithmetic and an Embedded C implementation provides only 16 bits because that is the native size of the processor. When the algorithm is expressed in Embedded C, it will not produce outputs of the right precision.

In such a case, there is a mismatch between the requirements of the application and the capabilities of the processor. Under no circumstances, including the use of assembly, will the algorithm run efficiently on such a processor. Embedded C cannot overcome such discrepancies. Yet, Embedded C provides a great improvement in the portability and software engineering of embedded applications. Despite many differences between performance-

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Special Issue 2, March 2019

specific processors, there is a remarkable similarity in the special-purpose features that they provide to speed up applications.

Writing C code with the low-level processor-specific support may at first appear to have many of the portability problems usually associated with assembly code. In the limited experience with porting applications that use Embedded C extensions, an automotive engine controller application (about 8000 lines of source) was ported from the eTPU, a 24-bit special-purpose processor, to a general-purpose 8-bit Freescale 68S08 with about a screen full of definitions put into a single header file. The porting process was much easier than expected. For example, variables that had been implemented on the processor registers were ported to unqualified memory in the general-purpose microprocessor by changing the definitions in the header definition and without any actual code modifications. The exercise was to identify the porting issues and it is clear that the performance of the special-purpose processor is significantly higher than the general-purpose target.

2.3 ARDUINO IDE

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

V. CONCLUSION

In this paper we have presented a jointly trained network for simultaneous car license plate detection and recognition. With this network, car license plates can be detected and recognized all at once in a single forward pass, with both high accuracy and efficiency. By sharing convolutional features with both detection and recognition network, the model size decreases largely. The whole network can be trained approximately end-to-end, without intermediate processing like image cropping or character separation. Comprehensive evaluation and comparison on three datasets with different approaches validate the advantage of our method. In the future, we will extend our network to multi-oriented car license plates. In addition, with the time analysis, it is found that NMS takes about half of the whole processing time. Hence, we will optimize NMS to accelerate the processing speed.

REFERENCES

1. L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in Proc. Asian Conf. Comput. Vis., 2011, pp. 770–783.
2. L. Neumann and J. Matas, "Real-time scene text localization and recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3538–3545.
3. K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in Proc. IEEE Int. Conf. Comput. Vis., Nov. 2011, pp. 1457–1464. This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination. LI et al.: TOWARD END-TO-END CAR LICENSE PLATE DETECTION AND RECOGNITION 11
4. T. Wang, D. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in Proc. IEEE Int. Conf. Pattern Recognit., Nov. 2012, pp. 3304–3308.
5. A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 785–792.
6. M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in Proc. Eur. Conf. Comput. Vis., 2014, pp. 512–528.
7. M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," Int. J. Comput. Vis., vol. 116, no. 1, pp. 1–20, 2016.
8. M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in Proc. Nat. Conf. Artif. Intell., 2017, pp. 4161–4167.
9. S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 311–325, Feb. 2013.
10. W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," IEEE Trans. Image Process., vol. 21, no. 9, pp. 4269–4279, Sep. 2012.
11. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," IEEE Trans. Intell. Transp. Syst., vol. 7, no. 3, pp. 377–392, Sep. 2006