ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

Security in Automotive Domain Using Secure Socket Layer

Alphonsa Johny, Jayasudha J.S, Anurag R

Abstract— Modern cars contain a million of lines of software codes, and if we bought a premium class automobile recently, it probably contains close to 100 million lines of software code. All that software executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of our car. Software in cars is only going to grow in both amount and complexity. This means that most new cars are executing tens of millions of lines of software code, controlling everything from brakes to the volume of the radio. A wide variety of communication systems are available in today's automotive network, for different applications range from body systems, engine control, driving assistances and safety systems to a wide variety of infotainment applications. Original equipment manufacturers (OEM) are spending lots of money and effort for developing software's that can provide the above said services and connectivity services. But today's tablets are highly developed and if we are successful in integrating tablets securely to the vehicle network, original equipment manufacturers can save a lot of money and effort. But with the integration of tablets, the secure car network is exposed to Internet which is susceptible to many vulnerabilities and attacks. Secure Tablet Integration Using Secure Socket Layer shall provide a secure and reliable connectivity within the car.

Index Terms—**Electronic Control Unit (ECU), Gateway, Original Equipment Manufacturer (OEM), Secure Socket Layer (SSL), Transport Layer Security (TLS).**

I. INTRODUCTION

The internet uses TCP/IP for data transfer and the TCP/IP protocol does not have an inherent security/protection mechanism. So the data in transition may get affected by active attacks such as message forgery and message alteration [1]. In order to protect the data in transmit, Netscape introduced the concept of SSL protocol in 1994 and later other companies such as Microsoft began to develop their own security protocols. Then Internet Engineering Task Force (IETF) intervened to define a standard for an encryption-layer protocol. With the input from multiple vendors, the IETF created Transport Layer Security standard. Previous versions of SSL are SSL 2.0 and SSL 3.0.Transport Layer Security (TLS) is a protocol based on SSL 3.0 and TLS 1.0 is same as SSL 3.1.Eventhough TLS and SSL protocols differ slightly in their implementation, the application developer and user cannot detect any differences at all [2]. Any network that needs to transmit data over an unsecured network such as Internet can make use of this SSL protocol to ensure security. Today SSL is widely used in online banking transactions for protecting the valuable user credentials such as password, PIN number etc, military purposes, embedded systems etc. It provides secure communication between the client and server. Here, the Client is our browser and server is the web server with which we are communicating [3]. A communication is said to be secure if it ensures confidentiality, authentication and integrity of the message [4]. Confidentiality is used to ensure that only authorized persons are reading the message. By integrity property, it means that the message is not modified in transmit. Authentication is achieved by using digital certificates and it is used to ensure that we are communicating with the real users and not to any imposter. In this paper there is a detailed description of SSL protocol [5]. Begin with the structure of SSL, its position in the network architecture, the later portion of this paper covers the various communication networks used in cars and an approach to provide secure vehicle communication using SSL.

II. SSL STRUCTURE

SSL comes in between the http application layer and TCP. Figure 1 shows the position of SSL in Network Architecture. SSL requires few changes to the protocols above and below it . The http application interfaces with SSL nearly in the same way it would with TCP in the absence of security. As far as TCP is concerned, SSL is just another application using its services. SSL is composed of two sub-protocols, the Hand-shake protocol and the Record protocol.

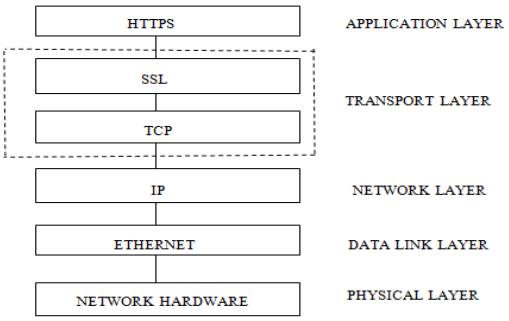

Fig 1 SSL Position in the Network Architecture

A. Hand-Shake Protocol

Hand-shake protocol is called so because it performs the initial handshaking operations such as certificates exchanging, key materials exchanging and identity authentication. The various step involved in the hand-shake protocol are given below [6].

- A client sends a *Client Hello* message to a server to ask for a SSL connection. This *Client Hello* message specifies the highest version of SSL it supports, a

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

random number, a list of suggested cipher suits and a compression method the client can support.

- The Server responds with a *Server Hello* message, containing the chosen protocol version, a random number, cipher suite, and compression method from the choices offered by the client. The server may also send a session ID as part of the message to perform a resumed handshake.
- The Server sends its *Certificate message* (depending on the selected cipher suite, this may be omitted by the Server).
- The Server sends a *ServerHelloDone* message, to indicate that handshake negotiation is completed.
- The Client responds with a *ClientKeyExchange* message, which may contain a pre-master secret, public key, or nothing. (It depends on the selected cipher.)
- The Client and Server use the random numbers and Pre-Master Secret to compute a common secret, called the *"master secret"* and then the both parties use the master secret to compute a key-block. All other key data for this connection is derived from this key-block (the client and server-generated random numbers), which is passed through a carefully designed *"pseudorandom function"*. The key data includes two session keys: *client_write_key* and *server_write_key*.
- The Client now sends a *ChangeCipherSpec* record, essentially telling the Server, "Everything I tell you from now on will be encrypted." The *ChangeCipherSpec* is itself a record-level protocol.
- Finally, the Client generates a *finished* message containing a hash and MAC over the previous handshake messages, and encrypts it using the *client_write_key* before sending it.
- The Server will attempt to decrypt the Client's *finished* message by the *client_write_key*, and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down.
- Finally, the Server sends a *ChangeCipherSpec* and its encrypted *finished* message, and the Client performs the same decryption and verification.

At this point, the "handshake" is complete and the application protocol is enabled. Application messages exchanged between Client and Server will be encrypted.

B. Record Protocol

The record protocol uses the session keys produced in hand-shake protocol to encapsulate the data to be exchanged. The encapsulation can provide confidentiality and integrity for data.

C. Cipher Suites

Before sending data, the sender and receiver must reach at a conclusion regarding the components of the selected cipher suite [7]. The four components of the cipher suite are key exchange algorithm; authentication technique, encryption method and the algorithm to compute the message digest hash. The key exchange algorithm specifies how the peers agree upon a common symmetric key that can be used to encrypt the message after the handshaking is done. The two common key exchange algorithms are DHE (Diffie-HellmanKeyExchange) and RSA (Rivest-Shamir-Adelman).The authentication algorithm indicates how the Client and Server will prove their identities to each other. Authentication options include RSA (Rivest-Shamir-Adelman), DSA (Digital signature algorithm), Elliptic curve DSA, PreShared Key (PSH) and anonymous (when no authentication mechanism is used) . Next, the encryption component indicates which symmetric algorithm is to be used to encrypt the data before transmission.RC4 (RivestCipher 4), DES (Data Encryption Standard), 3des (triple des), and AES (Advanced Encryption Standard) are the commonly used ones. The RC4 algorithm is the fastest and smallest of the cipher algorithms, and therefore is ideal for embedded processors. And finally the digest hash component is used to ensure that the receiver receives what the sender has transmitted, i.e. the message is not modified in transmission. SHA-1 is the most commonly used checksum algorithm to confirm the integrity of the exchanged data.

III. COMMON SSL LIBRARIES

The most popular implementation of TSL (Transport Layer Security) protocols is CyaSSL, OpenSSL, and MatrixSSL. The MatrixSSL itself comes in two versions-open and commercial. A brief description of each one of is given below.

A. OpenSSL

OpenSSL is based on the excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The main feature of OpenSSL Project is that it is full-featured and Open Source toolkit implementing the Secure Socket Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols [10]. The latest version of OpenSSL is OpenSSL 1.0.1c and it is licensed under an Apache-style license, and hence the developers are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions. OpenSSL toolkit can be used for the following purposes.

- Creation and management of private keys, public keys and parameters.
- Public key cryptographic operations.
- Creation of X.509 certificates, CSRs and CRLs.
- Calculation of Message Digests.
- Encryption and Decryption with Ciphers.
- SSL/TLS Client and Server Tests.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

- Handling of S/MIME (Multipurpose Internet Mail Extension) signed or encrypted mail.
- Time Stamp requests, generation and verification.

OpenSSL supports a number of different cryptographic algorithms: Ciphers supported by OpenSSL are AES(Advanced Encryption Standard),Blowfish, Camellia, SEED(It is a block cipher developed by Korean security Agency), CAST-128(alternatively known as CAST5,it is a block cipher gets its name from the inventors Carlisle Adams and Stafford Tavares),DES(Data Encryption Standard), IDEA,RC2(Rivest-Cipher 2) RC4, RC5, Triple-DES and GOST 28147-89. The following Cryptographic hash functions can be used with OpenSSL MD5 (Message Digest5), MD2, SHA-1(, SHA-2, RIPEMD-160, MDC-2, GOST R 34.11-94.

B. CyaSSL

The CyaSSL is an embedded SSL library, which is lightweight and written in ANSI C [11]. It is targeted for embedded and Real Time Operating System (RTOS) environments-mainly because of its small size, speed, and feature set. CyaSSL supports industry standards up to the current TLS 1.2 level and DTLS (Datagram Transport Layer Security). It is up to 20 times smaller than OpenSSL, and offers progressive ciphers such as HC-128, RABBIT, and NTRU .It offers full client and server support. OCSP and CRL support is also there in CyaSSL. Its size is in the range 30-100kB.Runtime memory requirement is 3-36kB.The first major user of CyaSSL/yaSSL was MySQL, the world's most popular open source database. CyaSSL is currently available for Win32/64, Linux, Mac OS X, Solaris, FreeBSD, NetBSD, OpenBSD, embedded Linux, Haiku, Open wrt,iPhone , Android, Nintendo Wii and GameCube through DevKitPro support, QNX, VxWorks, Monta Vista, ThreadX, Tron variants, OpenCL, Micrium'sMicroC-OS/II, FreeRTOS, Freescale MQX, and Nucleus.

C. MatrixSSL

MatrixSSL is an embedded SSL and TLS implementation designed for small footprint applications and devices [12]. It is available as an open source resource and it perfectly match with the requirements of embedded systems (memory constraint, processing capability and battery power constraints). MatrixSSL has been ported to operating systems including VxWorks, uClinux, eCos, FreeRTOS, ThreadX, ARM, MIPS32, PowerPC, H-8, SH3, i386 and x86-64.Important features of MatrixSSL are given below

- Total footprint of MatrixSSL with crypto provider is less than 50KB.
- SSL 3.0, TLS 1.0 and 1.1 servers and client support.
- Included crypto library - RSA, ECC, 3DES, AES, ARC4, SHA1, MD5, and RC2.
- Cipher Suites - RC4-MD5, RC4-SHA, DES-CBC3-SHA, AES128-SHA, AES256-SHA.
- Full support for session resumption/caching.
- Session re-keying and cipher renegotiation.
- Server and client X.509 certificate chain authentication.
- Parsing of X.509 pem and ASN.1 DER certificate formats.
- PKCS#1.5, PKCS#5 PKCS#8 and PKCS#12 support for key formatting.
- SSH command line support.
- Fully cross platform, portable code base; minimum use of system calls.
- Pluggable cipher suite interface.
- Pluggable crypto provider interface.
- Pluggable operating system and malloc interface.
- TCP/IP optional.
- Multithreading optional.
- Only a handful of external APIs, all non-blocking.
- Clean, heavily commented code in portable C.

IV. BUS REPRESENTATIVES

The various Electronic Control Units (ECU's) in the vehicle network are interconnected by bus network. Controller Area Network (CAN), Local Interconnect Network (LIN), FlexRay, Media Oriented System Transport and Bluetooth are the commonly used interconnection network in the in-vehicle communication system. A brief discussion of each one is given in the following section.

A. Controller Area Network

The all-round Controller Area Network, developed in the early 1980s, is an event-triggered controller network for serial communication with data rates up to 1 MBit/s. Its multi-master architecture allows redundant networks, which are able to operate even if some of their nodes are defect. CAN bus use the decentralized, reliable, priority driven CSMA/CD (Carrier Sense Multiple Access / Collision Detection) access control method to guarantee every time the transmission of the top priority message always first. In order to employ CAN also in the environment of strong electromagnetic fields, CAN offers an error mechanism that detects transfer errors, interrupts and indicates the erroneous transmissions with an error flag and initiates the retransmission of the affected message. Furthermore, it contains mechanisms for automatic fault localization including disconnection of the faulty controller [13]. CAN a link-layer data protocol—has become the dominant communication network for in-car networks (e.g., used by BMW, Ford, GM, Honda, and Volkswagen). A CAN packet does not include addresses in the traditional sense and instead supports a publish-and-subscribe communications model. The CAN ID header is used to indicate the packet type, and each packet is both physically and logically broadcast to all nodes, which then decide for themselves whether to process the packets or not.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

B. Local Interconnect Network

LIN (Local Interconnect Network) was developed as cost-effective alternate to CAN protocol. In 1998 a group of companies including Volvo, Motorola, Audi, BMW, Daimler Chrysler, and Volkswagen formed a consortium to develop LIN [14]. The LIN is a SCI/UART (Universal Asynchronous Receiver Transmitter) based serial, byte-oriented, time triggered communication protocol designed to support automotive networks in conjunction with Controller Area Network (CAN), which enables cost effective communication with sensors and actuators when all the features of CAN are not required. The main features of this protocol (compared to CAN) are low cost and low speed and used for short distance networks. Usually in automotive application, the LIN bus is connected between smart sensors or actuators and an Electronic Control Unit (ECU) which is often a gateway with CAN bus. Like CAN, LIN is also a broadcast type serial network, but with single master and multiple (up to 16) slaves. No collision detection exists in LIN, therefore all messages are initiated by the master with at most one slave replying for a given message identifier. The master is typically a moderately powerful microcontroller, whereas the slaves can be less powerful, cheaper micro-controllers. Moreover the LIN is a single wire 12V bus connection, in which the communication protocol is based upon ISO 9141 NRZ (Non return to Zero) standard. An important feature of LIN is the synchronization mechanism that allows the clock recovery by slave nodes without quartz or ceramics resonator. Only the master node will be using the oscillating device. Nodes can be added to the LIN network without requiring hardware or software changes in other slave nodes. And the maximum transmission speed will be 20kbit/s. It is intended to be used from the year 2001 on everywhere in a car, where the bandwidth and versatility of a CAN network is not required. A single master controls collision-free communication with up to 16 slaves, optionally including time synchronization for nodes without a stabilized time base. LIN is similarly to CAN a receiver-selective bus system. Incorrect transferred LIN messages are detected and discarded by the means of parity bits and a checksum. Beside the normal operation mode, LIN nodes provide also a sleep mode with lower power consumption, controlled by special sleep wake-up message.

C. FlexRay

FlexRay is a deterministic and error-tolerant high-speed bus, which meets the demands for future safety-relevant high-speed automotive networks [15]. With its data rate of up to 10 MBit/s (redundant single channel mode) FlexRay is targeting applications such as Drive-by-Wire and Power train. The flexible, expandable FlexRay network consists of up to 64, point-to-point or over a classical bus structure connected nodes. Both optical fibres and copper lines can be used as the physical transmission medium. FlexRay is similar to CAN a receiver-selective bus system and uses the cyclic TDMA (Time Division Multiple Access) method for the priority-driven control of asynchronous and synchronous transmission of non-time-critical respectively time-critical data via freely configurable, static and dynamic time segments. Its error tolerance is achieved by channel redundancy, a protocol checksum and an independent instance (bus guardian) that detects and handles logical errors.

D. Media Oriented System Transport

The ISO/OSI standardized MOST (Media Oriented System Transport) serial high-speed bus became the basis for present and future automotive multimedia networks for transmitting audio, video, voice and control data via fiber optic cables. The peer-to-peer network connects via plug-and-play up to 64 nodes in ring, star or bus topology. MOST offers, similarly to FlexRay, two freely configurable, static and dynamic time segments for the synchronous (up to 24 MBit/s) and asynchronous (up to 14 MBit/s) data transmission, as well as a small control channel. The control channel allows MOST devices to request and release one of the configurable 60 data channels. Unlike most automotive bus systems, MOST messages include always a clear sender and receiver address. Access control during synchronous and asynchronous transmission is realized via TDM (Time Division Multiplexing) and CSMA/CA respectively. The error management is handled by an internal MOST system service, which detects errors over parity bits, status flags and checksums and disconnects erroneous nodes if necessary.

E. Bluetooth

Originally developed to unify different technologies like computers and mobile phones. Bluetooth is a wireless radio data transmission standard in the license-free industrial, scientific and medical (ISM) band at 2.45 GHz. It enables wireless ad-hoc networking of various devices like personal digital assistants (PDAs), mobile phones, laptops, PCs, printers and digital cameras for transmitting voice and data over short distances up to 100 meters. Primarily designed as low-cost transceiver microchip with low power consumption, it reaches data rates of up to 0.7 MBit/s. Within the limited multi-master capable, so called Piconets, single Bluetooth devices can maintain up to seven point-to-point or point-to-multipoint connections.

V. REALIZATION OF SECURITY MECHANISM

A. Secure Gateway Module

To establish a secure connection between the in-car network and the external network (e.g.:- to a user's tablet) secure socket layer protocol can be used [16]. Secure Gateway Module must be there to monitor the communication between the vehicle network and the external network. We have used TMS570LS3137 microcontroller as the gateway Electronic Control Unit (ECU). This microcontroller is from Texas Instruments. The main features of TMS570LS3137 microcontroller is given below [17].

- High-Performance Automotive Grade Microcontroller for Safety Critical Applications.
- ARM Cortex– R4F 32-bit RISC CPU.
- Up to 180MHz System Clock.
- Core Supply Voltage (VCC): 1.2V nominal.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

- JTAG (*Joint Test Action Group)* Security Module.
- 96-channel Vectored Interrupt Module (VIM).
- IEEE 1149.1 JTAG, Boundary Scan and ARM Core Sight Components.
- Multiple Communication Interfaces such as 10/100 Mbps Ethernet Media Access Control (EMAC), FlexRay Controller with 2 channels.

The SSL server side code will run on this gateway module and SSL client module will run on the tablet. The Gateway module receives Controller Area Network (CAN) messages from the vehicle network and transmits it over the secure communication link to the tablet over Wi-Fi network. CyaSSL library is selected as the SSL library for this thesis work. CyaSSL is lightweight TLS/SSL library which can be used to provide security, authentication, integrity and confidentiality of the network communications over the embedded platform. It is best suited for the embedded system since it is 20 times smaller in footprint as compared to the OpenSSL library. The main features of CyaSSL are

- Requires only 30-100KB Flash Memory.
- Requires only 3-36kB RAM.
- Supports TLS 1, 1.1 and 1.2 (client and server).
- Supports Dynamic Transport Layer Security (client and server).
- Hashing functions supported by CyaSSL are MD2, MD4, MD5, SHA-1, SHA-2, SHA-256, SHA-512 and RIPEMD-160.
- CyaSSL supports both Block and stream ciphers. Supported ciphers include DES, 3DES, AES, ARC4, RABBIT and HC-128.
- Supports PEM (Privacy Enhanced Mail) and Certificates.

The figure 2 shows the protocol layers that are needed to run CyaSSL over the TMS570LS3731 micro controller. Detailed description of the software protocol layers is given below.

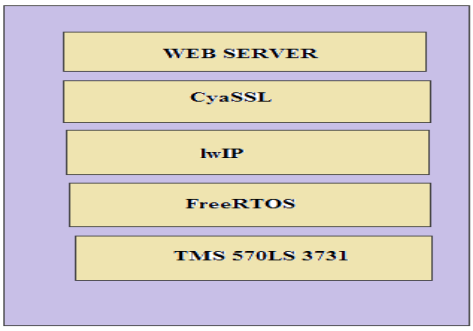

Fig 2 SSL over the Microcontroller Chip

A. Free Real Time Operating System (FreeRTOS)

A Real Time Operating System is an operating system that is optimized for use in embedded/real time applications. Their primary objective is to ensure a timely and deterministic response to events. FreeRTOS is a market leading real time operating system (RTOS). It is developed and supported by Real Time Engineers Ltd. It has a minimal RAM, ROM and processing overhead. Kernel binary image of the RTOS will be in the range of 4K to 9K bytes.

B. Light Weight Internet Protocol layer(LwIP)

The lwIP stack was originally developed by Adam Dunkels of the Networked Embedded Systems group at the Swedish Institute of Computer Science[18].The Lightweight IP (lwIP) stack is an open-source implementation of the TCP/IP stack developed specifically to reduce resource usage while maintaining a full-scale TCP/IP stack. For embedded systems, lwIP makes it possible to connect the system to a local intranet or the Internet. The lwIP stack has been ported to the Texas Instruments family of microcontrollers. LwIP can be implemented over both bare metal (system with no OS support) and systems with operating systems.LwIP are successfully ported to the FreeRTOS platform.

C. CyaSSL over FreeRTOS

CyaSSL is an application layer protocol and it requires a custom TCP/IP functionality. LwIP can be used to provide it. The following steps must be done for a TCP service to make use of the underlying FreeRTOS task.

- Create a new TCP Protocol Control Block(tcp_pcb)
- Bind it to IP_ADDR_ANY
- Put it in the listen mode
- Attach a tcp_accpet function to it and which then connects to the tcp_recv function once the connection gets established.
- In the receive function put the received data to a FreeRTOS queue.
- In a task which waits for this queue process the data and send back if needed.

Once the protocol stack is established that is FreeRTOS+lwIP, CyaSSL can run on the top of that and can provide secure communication between the client and the server. CyaSSL is delivered as a set of ANSI standard C source files and the required header files. The package can be easily added to the C project, and built with Code composer Studio compiler. In order to run CyaSSL over FreeRTOS and lwIP, it is required to define CyaSSL_FreeRTOS and CyaSSL_lwIP in /cyassl/ctaocrypt/settings.h.header file.

VI. SOFTWARE TOOLS USED

A. Code Composer Studio (CCSv5.2.1)

Code Composer Studio is the Integrated Development Environment (IDE) that can be used with microcontrollers from Texas Instruments. Code Composer Studio is based on the Eclipse open source software framework. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes compilers, source code editor, project build environment, debugger, profiler, simulators, real-time operating system and many other features. CCS is used in my thesis work to develop projects that can run on the TMS570LS3731 microcontroller.

B. GNU/Linux C++, C compiler

The secure socket layer server module which runs on the smart gateway is written in C programming language and the GNU/Linux C++ compiler is used to compile it.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

C. Eclipse Integrated Development Environment for Java Application development

Eclipse is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java. The secure socket layer proxy application and the vehicle test applications running on the tablet are written in Java programming language and Eclipse IDE is used to develop these applications.

REFERENCES

[1] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[2] S.Thomas, "SSL and TLS essential", John Wiley & Sons, inc, 2000.

[3] W.B.Mao,"Modern Cryptography: Theory and Practice," Publishing House of Electronics Indudtry, Beijing, 2004.

[4] G. Apostolopoulos, V. Peris V,D. Saha,"Transport Layer Security How much does it really cost", Infocom'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, IEEE, 1999, pp. 717-725.

[5] Alphonsa Johny, Dr. Jayasudha J.S,"Secure Socket Layer Implementations-A Review", International Journal of Computer Science & Engineering Technology (IJCSET),February 2013

[6] David Wagner, Bruce Schneier, "Analysis of the SSL 3.0 protocol", USENIX Workshop on ElectronicCommerce, ACM.

[7] Korea Information Security Agency, "A development of modules for Improving an ability and security of SSL/TLS", January, 2002.

[8] www.jucs.org."Embedded Monitors for Cryptographic Protocol Intrusion Detection and Prevention ", http://www.jucs.org/jucs_11_1/protomon_embedded_monitors_for/Joglekar_S_P.html.

[9] The Open Source Toolkit for SSL/TLS .www.openssl.org.

[10] Embedded SSL Library for Applications, Devices, and the Cloud, http://www.yassl.com/yaSSL/Home.html.

[11] MatrixSSLSpecification,www.matrixssl.org

[12] M. Norifusa, ―Safe border as SSL VPN Uniquely Enables New Style of Business Communications by Connecting Corporate Intranets and the Internet Seamlessly,‖ NEC Journal of Advanced Technology, vol. 1, April 2006, pp. 315-321.

[13] E. A. Young and T. J. Hudson," OpenSSL: The Open Source toolkit for SSL/TLS", http://www.openssl.org/.

[14] Z. Z. Bai, B. Q. Luo, and J. B. Xia, "Security realization of embedded Web Server based on SSL", Electronics Optics & Control, vol. 13,June 2006, pp. 60-62 .

[15] James Joy and Anurag R. "Architecture for secure Tablet integration in automotive network", Proceedings of the FISITA 2012 World Automotive Congress, Volume 6: Vehicle Electronics, SAE-China, 2012.

[16] TMS570 Transportation and Safety Microcontrollers, http://www.ti.com/lit/ml/sprt609/sprt609.pdf.

[17] Pierre Kleberger, Tomas Olovsson, and Erland Jonsson,"Security Aspects of the In Vehicle Network in the Connected Car", IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, June 5-9, 2011.

AUTHOR'S PROFILE

First Author Alphonsa Johny,M.tech Student, Department of Computer Science &Engineering, Sree Chitra Thirunal College of Engineering, Trivandrum, Kerla India.

Second Author Dr. Jayasudha J.S, Head of the Department, Department of Computer Science &Engineering, Sree Chitra Thirunal College of Engineering, Trivandrum, India.

Third Author Anurag R, Competency Manager, Tata Elxsi Ltd,Technopark, Trivandrum, India.