Secured and Lightweight Communication Scheme on UDP for Low End IoT Devices

Rohan A. Nathi, Embedded Engineer,
S.N. Systems Pvt. Ltd, Pune, India
rohan.nathi@gmail.com,

Dimpal Sutar, Embedded Engineer,
Zhypility Technologies Pvt. Ltd, Mumbai, India
dimpalssutar@gmail.com

Abstract—to consummate the demand of real time dispersal of sensor data over the Internet, researchers have wisely chosen UDP as a Transport layer for IoT communication. IoT industries do have potential to provide smart services, but these services are compromising the data security and privacy because of resource-constrained IoT environment. The security standards of the information being shared are a vital, open research issue. Our work aims to explore the UDP layer for IoT applications and designing a scalable, lightweight and secure communication scheme on UDP transport layer. This scheme will aid IoT product developers in designing an efficient, reliable and secured end to end IoT application. This paper implements and evaluates the real-time performance of the proposed scheme.

Keywords—Internet of Things(IoT), Datagram Transport Layer Security (DTLS), User Datagram Protocol (UDP),Constrained Resources, Global System for Mobile communication(GSM), General Packet Radio Service (GPRS);

I. Introduction

IoT has rooted into many applications from remote monitoring wireless sensor network to industrial applications. A forecast from experts [1] predicts that around 20.4 billion IoT devices will be connecting to the internet network by 2020 and hence privacy of information is at risk. IoT devices have constrained resources [2] in terms of memory, power and computational power and hence device management is always a challenging task. The multi-layered architecture comprising sensing/actuator, network and cloud applications over tiny footprint ECU, makes it a challenging research environment in order to implement security measures and device management.

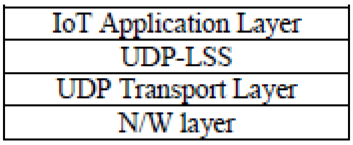

Figure 1: Network Model

Sensing/actuator layer consists of IoT device management, where data need to be collected from various sensors and processed for sending it to the cloud application. Wireless sensor network consists of tiny embedded devices connected to sensors and actuators which are responsible for controlling and sensing action. Network layer is responsible for routing the data to cloud application through the internet using various protocols. Cloud Application service layer is responsible for various components- vital two of them are web end (front end GUI) and database management for the IoT applications. More detailed overview of IoT with respect to enabling technologies, IoT architecture, and security and privacy issues has been given in work [3].

Secured and reliable end to end information delivery between the IoT devices and web front end interface across the Internet is a crucial task in the design phase of any IoT application. Swift developments in 5G, NB-IoT and CAT-M technologies, will pave the way for IoT devices in efficient utilization of bandwidth. Due to technological advancements, IoT embedded devices are mass produced and hence IoT implementations have become cheaper. Term “Massive IoT” describes a technological revolution where billions of IoT devices will be communicating with each other. To build such a huge scale infrastructure all the IoT application needs to efficient, scalable, and lightweight and secure [4]. TCP is a connection oriented transport protocol with mechanism that ensures the delivery of the data. TCP incurs huge headers (up to 60Bytes), which has special fields that guarantees the reliability of the data transfer. Complexity and computational states involved in a TCP transfer can prove burden to constrained resourced IoT devices. UDP is a connectionless and more efficient transport layer protocol that has tiny headers (8 bytes). It does not involve complex mechanism for delivery assurance and hence the reliability of data transfer is tampered. Reliability and efficiency of these two protocols are always contradictory challenge for IoT implementations [5].

Security, privacy and data integrity of the information for massive IoT infrastructure is the critical and the most crucial challenge for the researchers. Different security issues and potential threats like Man in-the-middle Attack (MITM), Rogue access point, and Device cloning attack for an IoT environment have been discussed by author Jean Pierre Nzabahimana [6]. The authors in work [7] had given detailed analysis of security issues and threat models from IoT device, Network and Cloud application perspective. Low end tiny IoT device has many hardware and software limitations which need to be addressed during the design phase of an IoT application [8]. Transport Layer Security (TLS), Secure Socket Layer (SSL) and Datagram Transport Layer Security (DTLS) are the widely adopted security communication protocols running over TCP and UDP over the internet. Application layer uses these protocols to establish an end to end secure communication channel between two entities over the internet using various symmetric and asymmetric encryption techniques. These protocols involves heavy handshaking mechanism and complex encryption algorithms and hence integrating these protocols into tiny embedded IoT devices would not be technically a right choice as it will overburden processing units[9].

We have devised a secured and lightweight communication scheme on UDP (henceforth UDP-LSS) for low end IoT devices considering constrained resources. This scheme involves pre-sharing of cryptographic embedded material at the deployment phase of devices. Symmetric encryption algorithms are used to secure the communications. The paper is divided into following sections- Sections II discusses the related work in UDP transport layer and other lightweight schemes. Section III presents proposed lightweight security scheme and its performance in real time environment with security analysis is evaluated in Section IV.

II. Background

To exploit the small header size in UDP, researchers have introduced many reliable communication schemes. A reliable scheme proposed in work [10] uses a queuing mechanism, where sender sends the packet to the queue, while the queue management software manages the entire queuing mechanism and the corresponding acknowledgement for each data packet sent out of the queue. Connection phase of this scheme uses the 3 way TCP handshaking mechanism to ensure connection oriented reliable service. Authors in work [11] has evaluated various reliable UDP communication schemes and concluded performance adaptive UDP showed the best result in terms of CPU utilization. DTLS is hefty protocol which is not suitable for IoT environment. It uses underlying UDP transport and three other sub protocols (handshake, alert and change cipher spec protocols) layer to establish a secure communication channel. Researcher has worked around in creating lightweight version of DTLS for IoT environment [12, 13, and 14], one of the work [14] has proposed Enhanced DTLS (E-Lithe) where authors have introduced trusted third party concept and pre-exchange of secret keys.

Work [15] formulates a lightweight encryption scheme based of identity based encryption technique. This scheme has devised flexible key management, where the keys are generated based on identity of device (for example MAC ID, Memory embedded serial number, etc.). Cryptographic algorithm over embedded platform need to be optimized as it consumes a significant amount of memory, computational power and battery resources. Susha Surendran ET. Al [16] has presented a comparison of various lightweight encryption algorithms, potential attacks on it and emphasized the need of light-weight ciphers in embedded platform for IoT industrial applications.

Motivation: IoT industry would deploy devices in massive scale for various applications, predominantly all these devices will be configured to communicate with trusted party server application; hence the proposed scheme will aid practitioners in designing and building lightweight, reliable and secured IoT applications.

III. Proposed Security Scheme

UDP-LSS is designed for a reliable end to end message delivery, with a low sized header that incorporates security mechanisms. The Scheme uses the Update-Response communication paradigm, where every message transaction/delivery is acknowledged with a response. Considering the constrained characteristics of IoT devices, we have also formulated a lightweight encryption algorithm for secure data transfer. UDP-LSS involves two entities which are as follows:

1) Application Server (AS) running IoT services to cater devices, which is hosted by a trusted third party. These services are responsible for security mechanism, inserting/updating received message as a record into the application server's database and managing GUI interfaces and IoT data statistics.

2) To enable lightweight secured communication with the application server, cryptographic components are embedded into the device's memory.

Further sections describe the details of the implementation of this scheme.

A. Cipher Key Management

An integrated key management mechanism has been adopted for UDP-LSS. Cipher Key definitions used in this scheme are as follows:

1) *ClientID (CID):* Each IoT device is been uniquely identified by server application by ClientID. It is a numerical string literal of fixed size.

2*) Secret Pre-shared Cipher Key (SPK):* SPK is 256 bit cipher key unique to each device.

3) *Tokens:* Token is insensitive string literal which is appended with every packet defined for this scheme.

4) *Token Cipher Key (TCK):* Each token has an associated 256 bit cipher key.

5) *Final Cipher Key (FCK):* This key is produced with combination of SPK and TCK using a special function devised for generation of this key, packet gets encrypted with the FCK.

During the provisioning and deployment phase of IoT device following embedded cryptographic components are exchanged between IoT device and Server Application. 1) Bucket of token and corresponding token keys. 2) Client ID (CID) and corresponding Secret Pre-shared Cipher Key (SPK). These components are securely embedded into device's memory and Application Server Database.

B. UDP-LSS Packet Definition:

UDP-LSS defines 5 types of packet. Header field of packets are designed in such manner which will have inbuilt security mechanism for protecting sensitive information from intruders. PACKET-TYPE is a 1 byte field that defines packet type. Every packet has timestamp field of 12 bytes, which includes device/server time (HHMMSS format) and date (DDMMMYY format). CRC field of 2 bytes is introduced which takes care of Transmission Errors. Packets are described as follows:

1) Heartbeat Packet (Packet Type - 1): This packet is sent by the device to Application Server. Fig shows the Heartbeat packet.

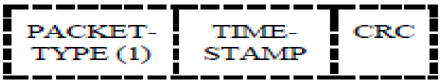

Figure 2: Heartbeat Packet structure

2) *Data Update Packet (Packet Type - 2):* This Packet is used transfer Sensor Data/Measured parameters to the Application Server; LEN field specifies the length of Data field.

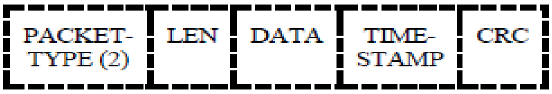

Figure 3: Data Update Packet structure

3) *Command Packet (Packet Type - 3):* Whenever any action is to be initiated by IoT device, this packet is sent by server application to IoT device. Data field specifies what action needs to be performed.

PACKET-TYPE (3)	LEN	DATA	TIME-STAMP	CRC

Figure 4: Command Packet structure

4) *Alarm Packet (Packet Type-4):* Any case of any abnormal condition, this packet is sent by IoT device to Application Server.

PACKET-TYPE (4)	LEN	DATA	TIME-STAMP	CRC

Figure 5: Alarm Packet structure

5) *Response-Acknowledgement Packet (PacketType-5):* Response-Acknowledgement Packet is sent as response to Alarm, Heartbeat, Data Update and Command Packet. A-STATUS (Authentication Status) is 1 byte field can have the following response code; 0x01:- Client is authenticated, 0x02:- Server is authenticated & 0x03:- Unauthorized client/server.

R-Code (Response-Code) is one byte field can have the following response code. 0x00:- Reception and Decryption of Message successful, 0x01:- Execution of Command successful, 0x02:- Retransmission Request (Error in Decryption of Message), 0x03:-Terminate the Session, 0x04:- Reserved for Future Scope.

PACKET-TYPE (5)	R-CODE	A-STATUS	TIME-STAMP	CRC

Figure 6: Response Packet structure

C. Formulating Cryptographic Algorithm and Embedding Security Mechanism

To enable lightweight computations for constrained environment, symmetric key encryption algorithm is used, As shown in Figure 7, UDP-LSS packet are processed with a block size of 64 bit. We have constructed the following functions that perform encryption on UDP-LSS packets.

1) Obfuscate_CID (CID, TCK): This function produces obfuscated cipher text of the client ID. CID is encrypted using TCK.

2) Key_Generate (SPK, TCK): This function takes two arguments SPK and TCK and returns FCK. Here substitution–permutation (S-P) operations are performed in SCK and TCK through S-P network and then EX-ORed to produce FCK. Key generate function model is shown in figure 8.

3) Encrypt_Packet (UDP-LSS Packet, FCK): This function takes two arguments. Then it performs substitution–permutation (S-P) operation through a (S-P) network. This network can have many sub - rounds. Computed S-P cipher text is EX_OR (□) with FCK to produce the encrypted data. When the device is ready for the data transfers following steps are computed.

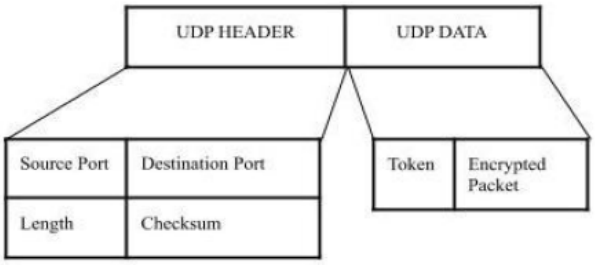

Figure 7: UDP packet structure

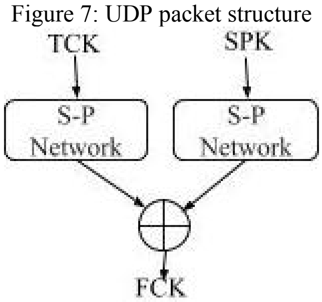

Figure 8: Key_Generate function Model

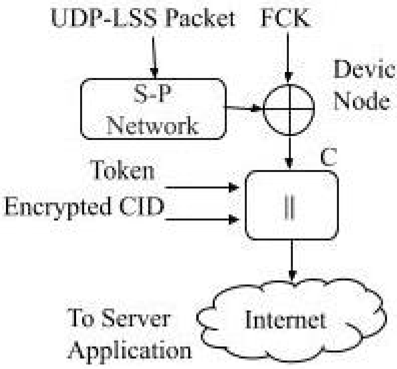

Figure 9: Encryption Model

i) The device selects a random token and its corresponding TCK, and then Key_Generate function is invoked to produce FCK and Obfuscate_CID function to produce encrypted CID.
ii) Then Encrypt_Packet function is invoked to produce cipher text (C).
iii) Finally randomly selected token and encrypted CID is appended with the encrypted cipher text data and sent to Application server over the Internet.

Here Scheme has adopted a timeout of 2seconds and the device will retransmit the packet. Encryption model is shown in figure 9. Application Server exactly performs the reverse operation to retrieve the data from the encrypted message received. Sequence of operation is as follows-
1) Token is extracted and corresponding TCK is found Application Server Database, if retrieving of TCK fails indicates illegal device accessing services, to which A-STATUS field of Response Packet is set to 0, response message is sent back and then UDP Session is terminated.
2) CID is retrieved from encrypted CID field using TCK; SPCK is retrieved from Application Server Database using

CID as query parameters. Key_Genrate function is invoked to produce FCK.

3) UDP-LSS packets are retrieved using FCK in exactly the reverse way as discussed in above.

4) Error Transmission is checked by computing CRC and validates against the received one.

5) Application Server then validates the received message time-stamp in a tolerable range comparing it with server's time-stamp.

If any error occurs in any of the operation sequence discussed above, the corresponding response code is sent back to the device. Time Computation Unit (T), viz is the time required for the message of the size (N) to be encrypted is defined as:-

$$T = K * N \text{ units}$$

Where K = No. of rounds the encryption algorithm engaged. In order to produce final cipher text message, (Figure 9) plain text is encrypted first using S-P Network and then second round using FCK, hence in our implementation K = 2. Time Complexity for the devised encryption algorithm is Ω (N) [21] and it is completely IoT hardware platform dependent.

D. UDP-LSS Protocol Phases

We have designed UDP-LSS for any generic IoT applications. After every successful boot up and network initialization of the device, it sends heartbeat packet to the Application Server which indicates that the device is now online and ready for data transfer. In response to Heartbeat packet, Application Server sends a response packet to the device with the respective response code. There are three scenarios in UDP-LSS which are discussed as follows:

1) Data Updating Phase: Any IoT application would have a chain of sensors, sensing and producing information which needs to be regularly updated to Application Server for monitoring, statistics and maintenance purpose. Data field will contain this information, for example JSON format is shown below.

{"Sensor1": 23.6, "Sensor2":25.5,
"Sensor3":5, "Sensor4":1002}

Data Update Packet serves this phase.

2) Alarm Phase: In case of any abnormal activity sensed by device, needs immediate attention by Application Server. Data field will contain the fault conditions. Consider Sensor 1 is tampered, while Sensor2 starts creating warnings of an outer limit readings, this condition can be represented in the JSON format as shown below:

{"Sensor1": "Tampered", "Sensor2": "Critical",
"Sensor3": "OK", "Sensor4": "OK"}

Alarm Packet serves this phase.

3) *Command Phase:* The application server wants to initiate an action, for example turning on/off an actuator, relay or any digital output then command packet is sent to the device. Application Server can respond to alarm packet by sending command packet, stating an action to clear alarm abnormality. Data field will contain the respective commands. JSON format example is shown below.

{ "Actuator1 ": "ON", "Actuator2": "OFF",
"RELAY": "ON", "DO": "OFF" }

Now in all the three phases Data field can be altered as per IoT application design, we have designed these phases for all the generic IoT applications.

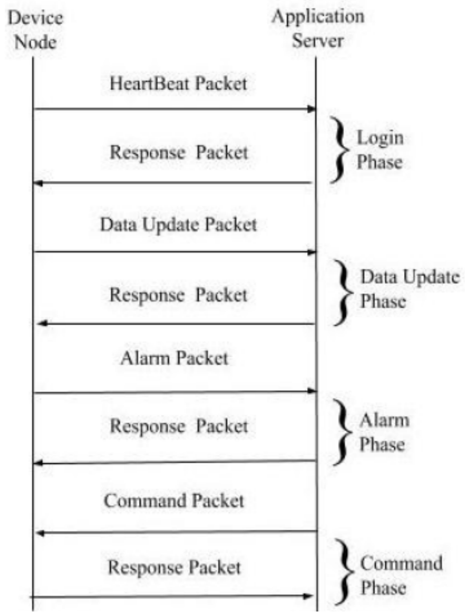

Figure 10: UDP-LSS Scheme

E. Use Case: Vehicle Tracking Device (VTS)

Vehicle tracking device is a low end tiny IoT device installed in vehicle. Device captures the navigational data that includes latitude, longitude, course and speed from various satellite systems (GPS, GNSS and IRNSS). This data is presented in a format described below and sent to server, the tracking application deployed on the server is responsible for processing this data and displaying the tracking and vehicle statistics on web frontend. This data produced by the device is vulnerable to intruders and can be easily tampered. Now to balance the tradeoff between the low-end IoT device and resources exhaustive security protocol proposed UDP-LSS protocol can be perfect solution. Data Update packet for Vehicle Tracking application is shown below[17], this packet is sent by the device to server at fixed interval of time to track the vehicle.

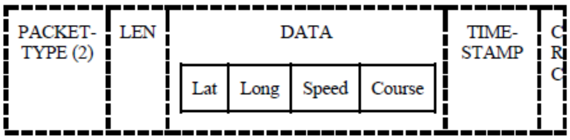

Figure 11: UDP-LSS Data Update Packet for VTS

IV. IMPLEMENTATION AND SECURITY ANALYSIS

The performance of UDP-LSS depends on the type hardware and software architecture engaged in designing the IoT application. IoT Devices can access the internet either through wired (Ethernet LAN) or wireless (Wi-Fi, GSM/GPRS) connectivity, and hence the characteristic of the service providing network to devices depends on the factors like network bandwidth, latency, congestion and traffic. Different IoT hardware device architectures implementing UDP-LSS will have different characteristics in terms of encryption and processing time. These characteristics will depend upon the hardware resources (Computing core, RAM, ROM). The real time performance of UDP-LSS is implemented and evaluated with the help of IoT device developed by R&D team at S.N. Systems Pvt. Ltd. This device is used for rapid prototyping in accomplishing various network related research for IoT environment. Hardware configuration of this device is shown in Table 1 below.

Table 1: Hardware Configurations for IoT Device

GSM-GPRS Module	SimCom 868 [18]. (Core:-ARM7, ROM:-512K, RAM:-448K and a small footprint of Linux based RTOS)
Power Supply	12V.
Data Transmission Medium	Mobile internet over 2G spectrum using General Packet Radio Services (GPRS)
Digital and Analog I/O	4 DI, 2 DO and 2 Analog I/P.

Node.js is scalable server side scripting language in runtime JavaScript open environment [19]. UDP-LSS is implemented in Node.js environment. It has various packages for network applications which can be easily implemented. UDP-LSS is implemented using NET package [20]. The application script is deployed on public server for real time performance evaluation. Table 2 shows server configuration.

Table 2: Server System Configurations

Operating System	64 bit, Windows 2008 Server.
Hardware	Intel(R)Xeon(CPU) X340 2.40 GHz
RAM	12 GB

IoT device had shown following observations:

1. Successful registration to GSM and GPRS network and UDP socket initialization took an average time of 20 seconds for 10 trails.

2. Cryptographic computation time is hardware architecture dependent; it will vary based on which hardware architecture is engaged. Table 3 shows the encryption timing for this device for data packet with different size.

3. Dummy data with different sizes have been sent to the application server, RTT is recorded after the response packet is received from the application server. Data sending frequency of 10 seconds for a time epoch of 5 min was recorded. Figure 11 shows the respective timings. (Note: All the RTT timings are dependent on the availability of internet and network traffic.)

Table 3: Encryption Time (in Millisecond)

Payload Size	50	100	200
Encryption time (MS)	1.040	1.600	2.150

A) Security analysis:

1) UDP-LSS Scenario with Man in the Middle Attack and Cloning Device Attack.

Trusted parties are involved in exchanging embedded cryptographic components between the device and the application server during deployment and provisioning phase. These components include keys described in the above section [3.A], which are stored securely with device's memory and application server's database. Each communication transaction over the channel is secured by generating FCK, hence every time a unique is generated. Therefore any alteration of payload is not possible by the hackers, without the knowledge of keys. Token is non-sensitive information, while CID is encrypted with token keys and hence it obsoletes cloning attacks.

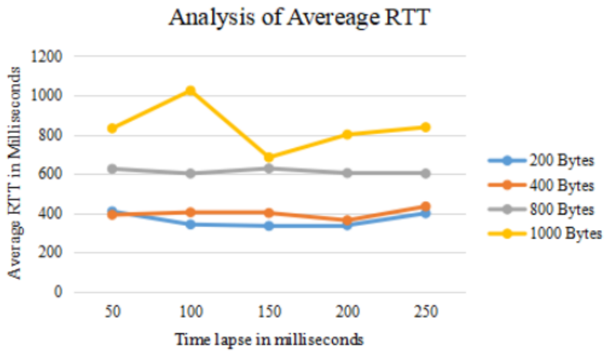

Figure 12 RTT Recordings

2) UDP-LSS Scenario with Replay Attack.

All the packets defined in UDP-LSS have a timestamp embedded into it. Validation of timestamp is computed by the server application by comparing with its own system timings in a tolerable range. Hence terminated sessions cannot be initiated, because the packet will have stale timestamp and it will be immediately rejected.

V. Conclusion

Information privacy and data integrity for IoT devices is very crucial aspect. Light-weight security scheme on UDP for IoT devices is being researched and implemented. The UDP-LSS aims to overcome the complex and highly resource consuming existing cryptographic algorithms, which is accomplished by introducing integrated key management during deployment phase. The complexity of key management in the proposed scheme can balance by lightweight feature of this scheme. The proposed scheme has defined small packets and 4 phases which can be adopted by any IoT applications, thus making communication simple. Scheme performance has been measured with RTT, encryption time. The further scope of the paper is to analyze its behavior on large scale in real-time applications.

Acknowledgment

This research was supported by S. N. Systems Pvt. Ltd and Zhypility Technologies Pvt. Ltd. We thank our colleagues who provided insight and expertise that greatly assisted the research.

References

[1] Gartner Inc Press Release, Available online at: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016.
[2] Z. Shelby, B. Frank, and D. Sturek,. "Constrained Application Protocol (CoAP). Internet-Draft" available online at: https://tools.ietf.org/html/rfc7252. Year: June 2014
[3] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications" in IEEE Internet of Things Journal, Year: 2017

[4] North stream White Paper, “Massive IoT: different technologies for different needs”, available online at: http://northstream.se/northstreamwp/wp-content/uploads/2017/06/Massive-IoT-different-technologies-for-different-needs.pdf".

[5] A. Milanovic, S. Srbljic, V. Sruk , “Performance of UDP and TCP Communication on Personal Computers” in 10th Mediterranean Electrotechnical Conference Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. Year:2000.

[6] Jean Pierre Nzabahimana, “Analysis of Security and Privacy Challenges in Internet of Things”. The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT Year: 2018.

[7] Md. Mahmud Hossain, Maziar Fotouhi, and Ragib Hasa “Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things” IEEE World Congress on Services, Year 2015.

[8] Subha Koley, Prasun Ghosal, “Addressing Hardware Security Challenges in Internet of Things: Recent Trends and Possible Solutions” in 12th IEEE International Conference on Ubiquitous Intelligence and Computing, Year: 2015.

[9] Iulia Florea, "Challenges In Security In Internet of Things" in 16th RoEduNet Conference: Networking in Education and Research (RoEduNet), Year: 2016.

[10] Xinhong HEI, Jia CHEN, Hongtao LU, Guo XIE, Haining MENG:“A UDP-based way to improve data transmission reliability” 29th Chinese Control And Decision Conference (CCDC), Year: 2017.

[11] Madzirin Masirap, Mohd Harith Amaran, Yusnani Mohd Yussoff, Ruhani Ab Rahman and Habibah Hashim, “Evaluation of Reliable UDP-Based Transport Protocols for Internet of Things (IoT)” in IEEE Symposium on Computer Applications & Industrial Electronics, Year: 2016

[12] Harshal Sardeshmukh, Dayanand Ambawade, “A DTLS based lightweight authentication scheme using symmetric keys for Internet of Things” in International Conference on Wireless Communications, Signal Processing and Networking,Year:2017.

[13] Vishwas Lakkundi, Keval Singh, “Lightweight DTLS implementation in CoAP-based Internet of Things” in 20th Annual International Conference on Advanced Computing and Communications (ADCOM), Year:2014.

[14] Asma Haroon, Sana Akram, Munam Ali Shah, Abdul Wahid, “E-Lithe: A Lightweight Secure DTLS for IoT ” in IEEE 86th Vehicular Technology Conference (VTC-Fall), year:2017.

[15] Sanaah Al Salami, Joonsang Baek, Khaled Salah, Ernesto Damiani, “Lightweight Encryption for Smart Home” in 11th International Conference on Availability, Reliability and Security (ARES), Year: 2016.

[16] Susha Surendran, Amira Nassef , Babak D. Beheshti “A Survey of Cryptographic Algorithms for IoT Devices” in IEEE Long Island Systems, Applications and Technology Conference (LISAT) Year: 2018.

[17] SeokJu Lee, Girma Tewolde , Jaerock Kwon, “Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application” in IEEE World Forum on Internet of Things (WF-IoT), Year:2014.

[18] SIMCom Wireless Solutions Ltd —SIM868 module, available online at: https://simcom.ee/modules/gsm-gprs-gnss/sim868/

[19] Documentation for Node.js environment, available online at: https://nodejs.org/en/docs/.

[20] Documentation for NET Node.js package, available online at: https://nodejs.org/api/net.html

[21] Thomas H. Cormen et. al (2013). *Introduction to Algorithms*. MIT Press