www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19296-19303

Adaptive Energy-Efficient SSL/TLS Method Using Fuzzy Logic for the MQTT-Based Internet of Things

Jin Hee Chung[1], Tae Ho Cho[2]
[1, 2] College of Information and Communication Engineering, Sungkyunkwan University,
Suwon 16419, Republic of Korea
{jinhee91, thcho}@skku.edu

Abstract: ***The Internet of things (IoT) has been undergoing accelerating growth, whereas the growth in its security is falling behind, representing a significant security problem. IoT devices use Message Queuing Telemetry Transport, which is a lightweight IoT standard messaging protocol, along with another security protocol. There are a few ways to secure communications, among which SSL/TLS is recommended. However, adapting SSL/TLS to IoT has many problems. First of all, the SSL/TLS handshake can incur considerable overhead, representing an intensive task for low-performance devices. Secondly, certain IoT communications use disposable handshakes, which are used with just one message when there is no need to connect in the future; this uses energy inefficiently and leads to reduced lifetime of the IoT devices. Thirdly, SSL/TLS may provide higher security levels than are needed for a transmission, in which case energy is wasted. Also, when only low security is required and the transmitted message is very short, SSL/TLS provides excessively secure communication and thus consumes energy inefficiently. Although there are the many problems, using SSL/TLS is unavoidable. Therefore, it is necessary to improve SSL/TLS for use with Message Queuing Telemetry Transport in the IoT. Herein we propose a method to improve energy efficiency in this environment while satisfying any of various expected security levels. Our proposed method recognizes the three inputs of security level, residual energy, and message length as the relevant device conditions. Based on these inputs, the proposed method selects an appropriate fuzzy rule to determine which client-compatible cipher suite is most suitable for the conditions. Herein we demonstrate the validity of our proposed method by means of experiments on its energy efficiency relative to that of SSL/TLS.***

Keywords: Internet of Things, Security, MQTT, SSL/TLS, Fuzzy Logic

1. Introduction

The Internet of things (IoT) is an infrastructure whereby many kinds of sensors and objects with embedded communication functions can be connected to the Internet [1]. IoT not only allows users to connect to and control such objects directly through the Internet, but also provides convenient and intelligent functions such as accurately sensing weather conditions from various devices in real time [2]. These days, improvements to IoT are accelerating and diverse IoT devices are already in use [3]. According to Gartner, an information technology research and advisory company, there are 22.9 billion IoT devices in 2016 and the number of IoT devices will increase to 50.1 billion by 2020. However, IoT security problems have allowed serious damages that diminish the potential benefits of intelligent devices. The estimated financial damage was 13.4 trillion in 2015, and notably IoT security cameras were recently eavesdropped in the USA [4]. Experts say that IoT security is a time bomb because the speed of progress in IoT is outpacing improvements in security. IoT objects range from high-performance devices such as computers and smartphones to low-performance devices such small office supplies. In the case of high-performance devices, some have energy constraints and some do not. However, nearly all low-performance devices have constraints on both computing ability and energy. This situation can lead to various IoT security problems if low-security methods are used because of the difficulties. As IoT technology improves, the types of information treated in IoT are expanding to encompass more critical applications, and the corresponding potential for damage is becoming enormous. Therefore, a method of low-energy, secure communications for IoT devices is needed.

IoT devices communicate by means [5] of low-energy protocols; the standard protocol is Message Queuing Telemetry Transport (MQTT) [6], a lightweight message transport protocol that transmits messages as plaintext [7]. Thus, MQTT should be used in tandem with a security protocol; SSL/TLS is recommended for this purpose [8]. However, SSL/TLS is not presently suitable for use with low-performance and low-energy devices for various reasons. Firstly, devices incur 6.5 KB average overhead to carry out a single handshake [9]; this represents an intensive task for low-performance devices [10]. When we consider the Arduino platform as an example, this 6.5 KB overhead is significant relative to the memory sizes of the microcontrollers: 2 KB for the Atmega328, 2.5 KB for the Atmega32U4, and 8 KB for the Atmega2560. Secondly, abusing handshake can reduce device lifetime. For instance, handshakes for disposable communication, which have low security requirements, or handshakes for very short messages consume energy inefficiently. Thirdly, even in the case of high-performance devices, many of these also have energy constraints and thus efficient energy consumption is still needed. SSL/TLS provides a high level of communication security that is often excessive, and thus also consumes excessive energy. Additionally, even in the case of a high-performance IoT device, the device may have low residual energy and thus have need to secure communication, even though the computational burden of high-security communication would not pose any problem.

Herein we propose a method to satisfy the minimum security requirements of IoT devices using MQTT while also consuming energy efficiently. Our method classifies IoT devices into five different groups according to communication purpose and function. We establish minimum security levels for each group. Furthermore, our method uses fuzzy logic to select a cipher suite suitable for the resources and security requirements of a device, taking into account the required security level, residual energy, and message length.

Section 2 of this paper gives background information and discusses related works, and Section 3 describes our proposed method in detail. Section 4 presents experimental results and analysis, and Section 5 gives our conclusions.

DOI: 10.18535/ijecs/v5i12.04

2. Background and Related works

2.1 MQTT

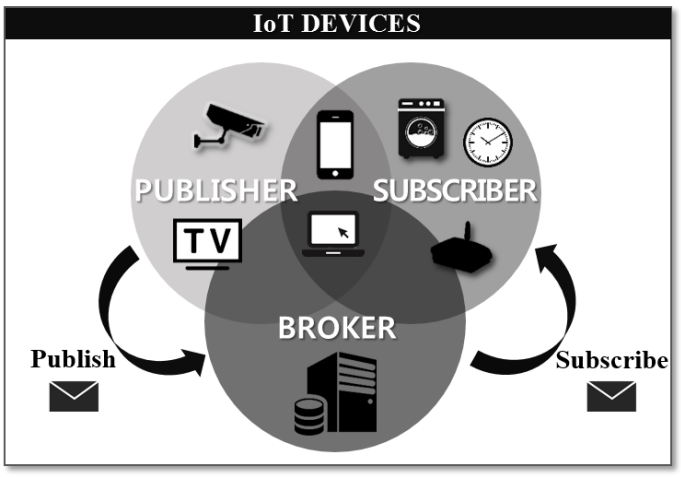

Figure 1: MQTT

MQTT was first developed by IBM and Arcom in 1999 and designated as the standard protocol for lightweight IoT messaging [8]. MQTT classifies IoT devices as Brokers, Publishers, and Subscribers (Fig. 1) [11]. A Broker is an intermediary that receives and transmits messages between Publishers and Subscribers. Publishers are devices that publish messages by transmitting them along with their topics to Brokers. Subscribers represent devices that subscribe to at least one topic and receive messages on these topics from Brokers. More than one device can publish and subscribe the topics. Also, many devices can subscribe one topic. This system is operated by topic and can be implemented hierarchically using slashes as topic level separators (Fig. 2).

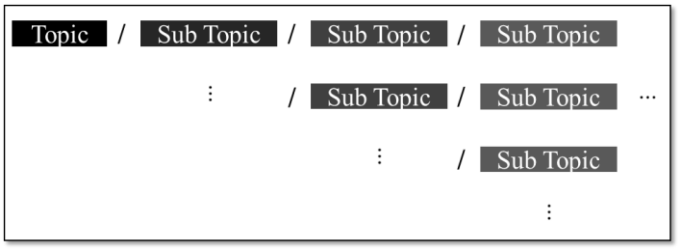

Figure 2: Hierarchy of MQTT Topics

Thus, many devices can be managed efficiently. Additionally, MQTT specifies three quality of service (QoS) levels that are intended to support reliable message transmission [12]. Because MQTT transports messages as plaintext, its security should be supplemented by some means; SSL/TLS is recommended for this purpose.

2.2 SSL/TLS

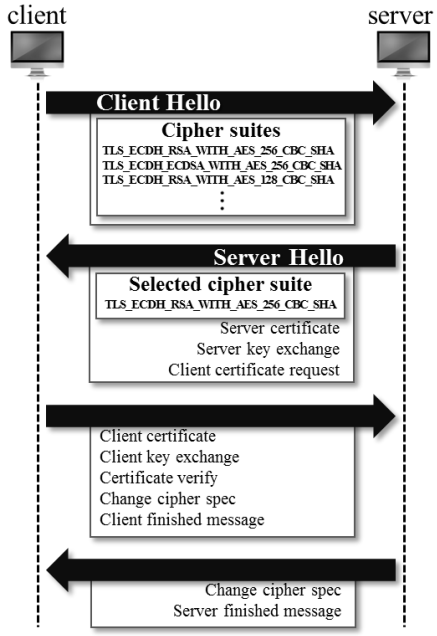

Figure 3: Handshake Process of SSL/TLS

SSL/TLS was first developed by Netscape and is a security protocol applicable to the TCP/IP network architecture [13]. SSL/TLS is an IETF standard regulation applied in diverse fields such as web browsing, electronic mail, and instant messaging to protect communications between a client and a server from eavesdropping, interference, and modulation. Most notably, it can protect from man-in-the-middle attacks and eavesdropping attempts that are made by means of packet spoofing, and can also authenticate the opposite party. Fig. 3 shows the handshake process of SSL/TLS. The first step of the handshake is the exchange of Client Hello and Server Hello messages to negotiate a cipher suite that will be used in the security session.

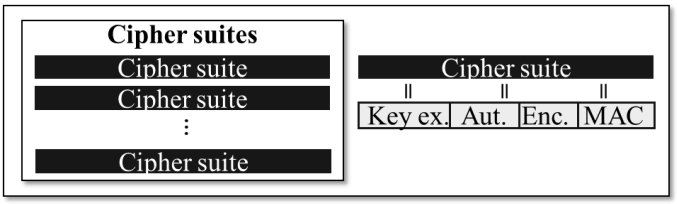

Figure 4: Structure of Cipher Suites

The Client sends a collection of the many cipher suites that it can support as part of its Client Hello, and the Server sends a selected cipher suite that can be supported by both the Client and the Server and has the highest preference as part of its Server Hello. The negotiated cipher suite includes ways to conduct key exchange, authentication, encryption, and MAC (Fig. 4) [14]. These items are not negotiated separately but rather as a group (i.e., a cipher suite). After the negotiation is complete, the selected cipher suite will be used for authentication and secure communication during the session.

2.3 Related Method to Improve Energy Efficiency

DOI: 10.18535/ijecs/v5i12.04

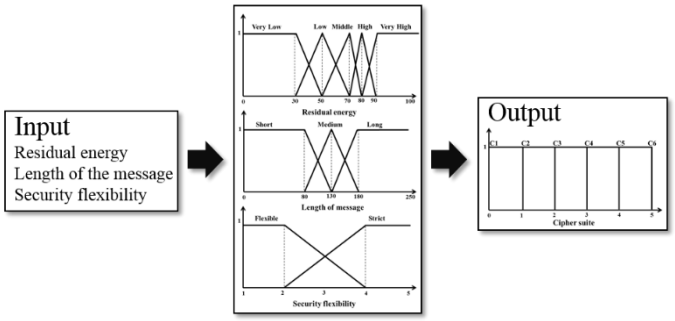

Figure 5: Fuzzy Process Proposed in a Related Work

Fig. 5 illustrates the fuzzy process used in a related work to improve the energy efficiency of IoT using MQTT with SSL/TLS [15]. In this process, the cipher suite is negotiated based upon the three inputs of residual energy, message length, and security flexibility. The security flexibility represents how flexible the need for security is in the communication. When it is Flexible, a low-energy and low-security cipher suite is selected; when it is Strict, a high-security cipher suite is selected regardless of its energy consumption.

3. Adaptive Energy-Efficient Method using Fuzzy Logic

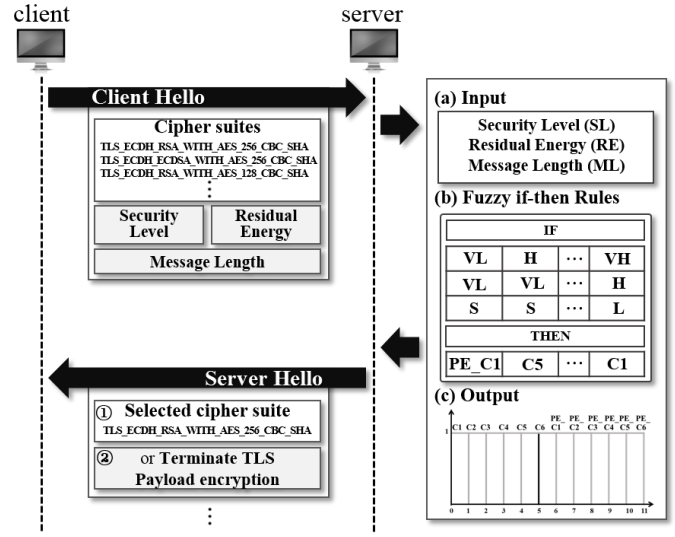

Figure 6: Process of the Proposed Method

Fig. 6 illustrates the process of security negotiation in the proposed method. The purpose of the proposed method is to improve energy efficiency, satisfy minimum security requirements, and maximize the security level within the given constraints. In this method, the Client Hello message representing the first step of the handshake includes the security level, residual energy, and message length. The information transmitted is as follows:

1) Security Level (SL): The security level represents the minimum security requirements expected for the communication; it is selected based upon the performance or function of the device. The SL should be satisfied and guaranteed. The specific security requirements at each level are detailed in Section 3.1.
2) Residual Energy (RE): The residual energy is the remaining energy of the device, represented as a percentage. It is used for saving energy and extending the lifetime of the device.
3) Message Length (ML): One of the most energy-intensive parts of using SSL/TLS is the encryption. Selecting an efficient cipher suite based upon the message length can save energy and extend the lifetime of the device. It supports message length of up to 256 MB.

After receiving this information, the server uses the inputs (Fig. 6a) and the fuzzy rule (Fig. 6b) to evaluate whether the client is able to communicate by means of SSL/TLS. The result of this process (Fig. 6c) determines how secure communication will proceed. If the client is determined to be able to communicate by means of SSL/TLS, a cipher suite is selected that satisfies the security requirements and consumes energy efficiently, and SSL/TLS with the selected cipher suite is initiated (Fig. 6, Step ①). If the client is determined to be unable to use SSL/TLS (Fig. 6, Step ②), a cipher suite is selected that satisfies the security requirements and consumes energy efficiently, SSL/TLS is terminated, and communication is conducted by means of a payload encryption process based upon the selected cipher suite (Fig. 7).

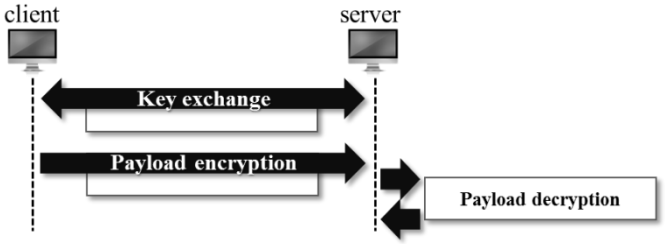

Figure 7: Payload Encryption Process

When the payload encryption process is selected, only the payload part of the MQTT message is encrypted using the key exchange algorithm and encryption algorithm of the selected cipher suite. This helps low-energy devices to conduct minimum-security communication using the minimum energy needed.

3.1 Security Levels and Security Requirements

Table 1: Security Levels and Security Requirements

Security Level	Purpose and function of the device	Minimum security requirements			
		Confidentiality	Integrity	Authentication	Non-repudiation
0					
≀	Public-purpose device with frequent disposable connections or broadcast purpose, under low security threat	o			
1					
≀	Connects indirectly to transport data only	o	o		
2					
≀	Is able to interact with, read, and write data	o	o	o	
3					
≀	Detects environmental information and converts it to a digital signal, or communicates with other devices and uses gateways (including sensing, actuating)	o	o	o	o
4					
≀	Has embedded processing and communication abilities (e.g. home appliances, smartphones)	o	o	o	o
5					

We establish the security requirements to satisfy the minimum security level and consider performance of the device. The security level is expressed as a rational number between zero and five, and it represents that the device has which purpose or function [16] (Table 1). For lower security levels, greater consideration is given to energy consumption and the security requirements are reduced. Accordingly, as the security level increases, less consideration is given to energy consumption and the security requirements are increased. The minimum security of our proposed method follows the newest TLS regulation (TLS v.1.2) by IETF [17]. In particular, it excludes RC4, IDEA, and DES as known vulnerable algorithms.

3.2 Fuzzy Logic System

The input parameters of the fuzzy logic system are as follows.

- Security Level (SL) = {VL (Very Low), L (Low), M (Medium), H (High), VH (Very High)}
- Residual Energy (RE) = {VL (Very Low), L (Low), M (Medium), H (High), VH (Very High)}
- Message Length (ML) = {S (Short), M (Medium), L (Long)}

The output parameter is the cipher suite:

- Cipher suite (C) = {C1, C2, C3, C4, C5, C6, PE_C1 (Payload Encryption_C1), PE_C2, PE_C3, PE_C4, PE_C5, PE_C6}

Table 2 lists the specific cipher suites.

3.2.1 Fuzzy Membership Function

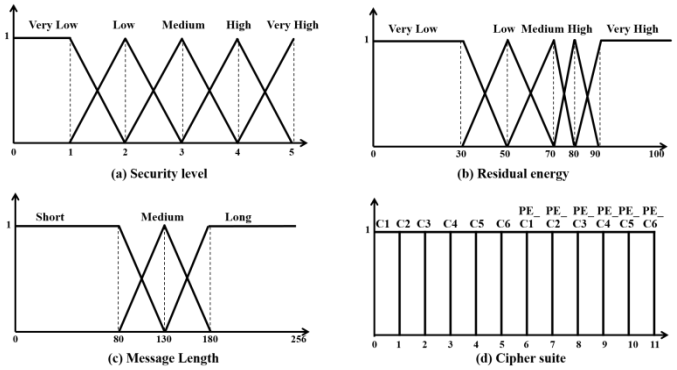

Figure 8: Fuzzy Membership Function

Fig. 8 illustrates the membership functions of the proposed method, including those for the three inputs (SL, RE, ML) and that of the output (C). The minimum and maximum values for each fuzzy set were chosen considering the conditions of IoT devices [18]. The membership degrees were established based upon data on the three input factors and then adjusted based on a tuning strategy for fuzzy membership functions and learned according to feedback from experimental performance [19]. Also, it uses a Mamdani-type inference method for aggregation and a center-of-gravity method for defuzzification [20].

3.2.2 Fuzzy Rules

In order to explain the fuzzy rules of our method, we use Table 2 as cipher suites which is supported by both client and server.

Table 2: Cipher Suites

No.	Cipher suite
C1	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
C2	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
C3	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
C4	TLS_DH_RSA_WITH_AES_128_CBC_SHA
C5	TLS_RSA_WITH_AES_256_CBC_SHA
C6	TLS_RSA_WITH_AES_128_CBC_SHA
C7	TLS_RSA_WITH_RC4_128_SHA

Our proposed method uses an evaluation function to calculate a score that is used to select a cipher suite suitable for the given conditions. Table 3 shows how the evaluation score (ES) is calculated for each entry in Table 2. Equation (1) is the function used to calculate ES.

Table 3: Evaluation Score of Each Cipher Suite

C \ Score	Energy consumption (E)	Preference (P)	Sum (S)	$\frac{E}{S} - \frac{P}{S} = ES$
C1	1	6	7	-0.71429
C2	2	5	7	-0.42857
C3	3	2	5	0.2
C4	4	3	7	0.142857
C5	5	4	9	0.111111
C6	6	1	7	0.714286
C7	X	X	X	X

$$\textit{Evaluation Score} = \frac{\textit{Energy consumption}}{\textit{Sum}} - \frac{\textit{Preference}}{\textit{Sum}} \quad (1)$$

As seen in Table 3, lower energy consumption (E) of a cipher suite score higher marks. Accordingly, higher preference (P) of a cipher suite score higher marks. After that, sum (S) obtains the sum of E and P. Lastly, it calculates evaluation function (ES) of each cipher suite. If a cipher suite does not satisfy the minimum security requirement, then it is ruled out like C7. Negative values of ES correspond to conditions of comparatively high preference and high energy consumption, whereas positive values of ES correspond to conditions of comparatively low preference and low energy consumption. Also, the absolute value of ES represents the relative gap between the preference and energy consumption. The final evaluation score is used for selecting a cipher suite, as explained in Table 4.

Table 4: Criteria for Establishing Fuzzy Rule

Security level	Residual energy	Message length	Output
VL			Payload encryption with a cipher suite having the highest P score
L		S	Payload encryption with a cipher suite having the highest P score (In this case, SSL/TLS overhead is too high because the message is short)
	VL		Payload encryption with a cipher suite having the highest P score
	Other		SSL/TLS communication with a cipher suite having the highest E score
M			SSL/TLS communication with the cipher suite having the ES closest to zero
H	For cipher suites X, Y, Z having the following trend of ES score: X < Y < Z		

	VL, L		SSL/TLS communication with Z
	M		SSL/TLS communication with Y
	VH, H		SSL/TLS communication with X
VH			SSL/TLS communication with a cipher suite having the highest P score

Table 5 shows a small part of established fuzzy rule. The fuzzy rule used depends on the cipher suites and the preference order of both the client and server. Thus, our proposed method gives a fuzzy rule adapted to the specific device conditions, rather than always using the same fuzzy rule.

Table 5: Fuzzy Rule

Rule No.	Input			Output
	SL	RE	ML	C
1	VL	VL	S	PE_C1
⋮				
20	L	VH	S	PE_C1
21	L	VL	M	PE_C1
22	L	L	M	C6
⋮				
30	L	VH	L	C6
31	L	VL	S	C5
32	M	L	S	C5
⋮				
46	H	VL	S	C5
47	H	L	S	C5
48	H	M	S	C2
⋮				
58	H	M	L	C2
59	H	H	L	C1
60	H	VH	L	C1
⋮				
75	VH	VH	L	C1

4. Experiment and Evaluation

To validate our proposed method, we compared its energy consumption with that of the original SSL/TLS method. Energy waste in general SSL/TLS communication can divided generally into encryption waste, including waste in the course of encryption/decryption, hashing, and electronic signature; and non-encryption waste, including waste in transmitting data or maintaining the network. In this experiment, we consider only encryption-related waste because non-encryption waste is outside the control of the proposed method.

4.1 Experimental Description

Table 6: Experimental Conditions

Number of trials	500
Security level	0~5
Residual energy	0~100%
Message length	0~256MB
Preference of cipher suite	Open SSL v.1.0.2
Cipher suites that are supported by both client and server	Table 2

Experiments were conducted to collect data for analysis that included the energy consumption of each cipher suite when used on a low-performance device [21]-[23], namely the Compaq iPAQ 3670 pocket PC (206 MHz CPU, 64 MB RAM). The preference order of cipher suites was established based upon the newest version (v.1.0.2) of OpenSSL. The experiment included 500 trials, each of which included the selection of random values for each of the three inputs (i.e. SL, RE, and ML), within the ranges listed in Table 6. The cipher suites used are listed in Table 2.

4.2 Evaluations

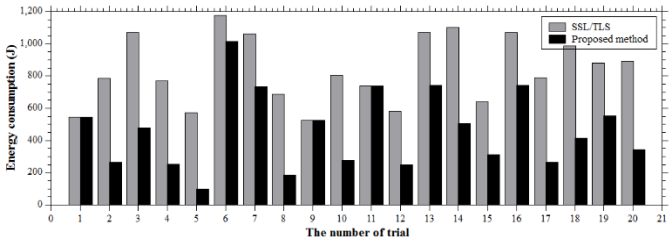

Figure 9: Energy Consumption Comparisons for 20 Trials

In the case of the original SSL/TLS method, suite C1 is always selected because it has the highest preference in the cipher suites. In all trials, the proposed method consumed energy equal to or less than that consumed by the SSL/TLS method. Among the 500 trials with random inputs, the energy efficiency was improved by an average of 43.54%. Fig. 9 shows the energy consumptions calculated for twenty trials of random inputs, for both the proposed method and SSL/TLS.

To analyze the efficiency of our method, we repeated the experiments with restricted ranges of values for each of the three inputs. First, to analyze the energy efficiency improvement of our proposed method for each security level, we conducted experiments within each of the five security levels (i.e. 0–1, 1–2, 2–3, 3–4, and 4–5). For instance, in the case of the 0–1 security level, random rational numbers between zero and one were generated and the RE and ML values were selected randomly within their entire ranges as listed in Table 6. Lower security levels, which correspond to lower security requirements and much simpler device purposes or functions, yielded the greatest energy efficiency improvements by means of the proposed method, relative to the use of the original SSL/TLS method (Figs. 10-12).

DOI: 10.18535/ijecs/v5i12.04

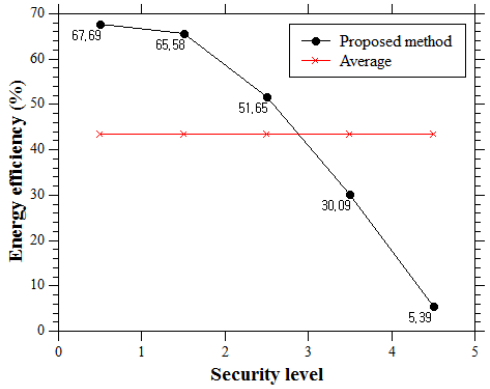

Figure 10: Energy Efficiency Improvements for Various Security Levels

Similarly, to analyze the effect of residual energy on the energy efficiency improvement of our method, we divided the range of residual energy into six subranges (0–30%, 30–50%, 50–70%, 70–80%, 80–90%, and 90–100) and repeated the experiment with random residual energy inputs within each of these, while selecting SL and ML values randomly within their entire ranges as listed in Table 6. The proposed method yielded the greatest improvements in energy efficiency for the RE range of 50–70% (Fig. 11).

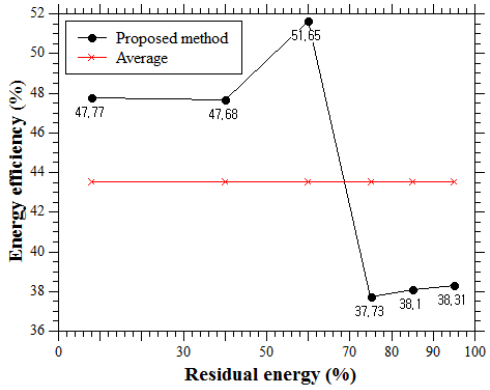

Figure 11: Energy Efficiency Improvements for Various Residual Energy

Lastly, to analyze the effects of message length on the energy efficiency improvement of our method, we divided the range of message length into four subranges (0–80, 80–130, 130–180, and 180–256 MB) and repeated the experiment for each subrange, generating random message lengths within each of these while selecting SL and RE values randomly within their entire ranges as listed in Table 6. Use of the proposed method yielded greater efficiency for shorter messages (Fig. 12).

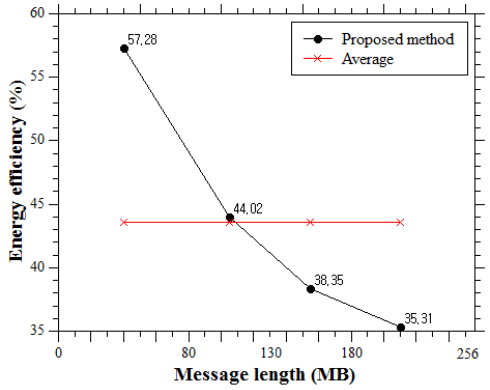

Figure 12: Energy Efficiency Improvements for Various Message Lengths

In summary, the experiments on the proposed method showed the following trends.

1) Lower required security levels yielded greater improvements in energy efficiency.
2) There was no straightforward relation between residual energy and energy efficiency.
3) The proposed method yielded greater energy efficiency improvements for shorter messages.

There are many reasons why these trends were observed. First, when our proposed method considers a cipher suite, it gives priority to security over residual energy. Thus, greater security requirements lead directly to poorer energy efficiency. Also, there are two reasons why there was no straightforward relation between residual energy and energy efficiency. Firstly, when there is little residual energy but the required security level is high, the residual energy problem is essentially ignored because maintaining the security level is a higher priority. Secondly, if the message is long, the computational burden is high even when the security level is low. Thus, in the case of residual energy, unlike the cases of varying security level or message length, there was no correlation. However, our method saved energy over the SSL/TLS method in all cases. Regarding message length, the reason why energy efficiency is better for shorter messages is that shorter messages require less computation. Also, for low security levels, the proposed method uses payload encryption instead of SSL/TLS for short messages, thereby saving a great deal of overhead.

We analyzed the worst case and best case found in the experimental results. Regarding the worst case, our method performs most poorly for security levels between four and five, residual energy between 70% and 80%, and message length between 180 to 256 MB. Fig. 13 shows the energy consumption of the proposed method and that of the SSL/TLS method for ten trials having all three input variables within their worst-case subranges. In the worst case, the energy consumption of our proposed method is equal to that of the original SSL/TLS method.

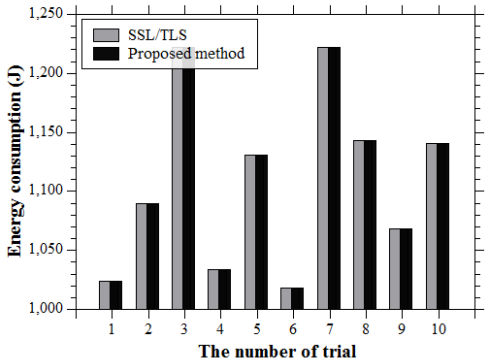

Figure 13: Worst-Case Performance of the Proposed Method

Regarding the best case, our method performs best for security levels between zero and one, residual energy between 50% and 70%, and message length between 0 to 80 MB.

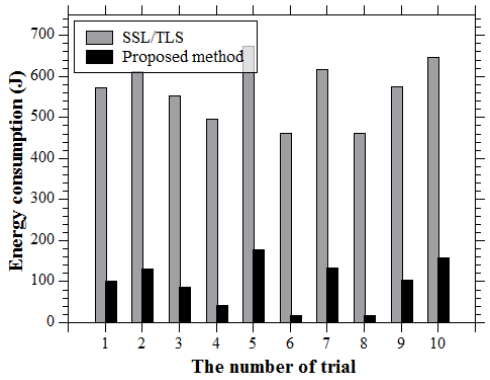

Figure 14: Best-Case Performance of the Proposed Method

Fig. 14 shows the energy consumption of the proposed method and that of the SSL/TLS method for ten trials having all three input variables within their best-case subranges. In the best case, the energy consumption of our proposed method saves on average 84.72% of the energy that would be used by the original SSL/TLS method.

5. Conclusions

The Internet of things is a technology that connects diverse kinds of devices of various performance levels. MQTT, a lightweight message transport protocol, is a standard protocol for the Internet of things. However, because it does not guarantee security, MQTT should be used in tandem with a security protocol. SSL/TLS is a key protocol among the security protocols available, but applying the original SSL/TLS to the Internet of things leads to several problems. SSL/TLS communication can represent an intensive task for low-performance and constrained devices, and its lack of energy efficiency can reduce device lifetime. As the number of the Internet of things devices is increasing every year, security problems are also increasing and the potential damages are significant. Therefore, improving SSL/TLS for Internet of things devices using MQTT is a meaningful endeavor. Herein we have proposed a method that adapts to specific device conditions by means of a fuzzy logic system. The proposed method selects an efficient cipher suite to improve energy efficiency given the device's needed security level. The device conditions considered in our method are the needed security level, residual energy, and message length. The purpose of our method to meet the minimum security requirements of the given security level and to improve the energy efficiency considering the residual energy and the message length. We demonstrated our method by means of experiments in which the energy efficiency was improved by 43.54% on average compared to the use of the original SSL/TLS method.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. NRF-2015R1D1A1A01059484)

References

[1] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey," Computer Networks, vol. 54, pp. 2787-2805, 2010.

[2] G. Kortuem, F. Kawsar, V. Sundramoorthy and D. Fitton, "Smart objects as building blocks for the internet of things," IEEE Internet Comput., vol. 14, pp. 44-51, 2010.

[3] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Comput. Syst., vol. 29, pp. 1645-1660, 2013.

[4] K. Zhao and L. Ge, "A survey on the internet of things security," in Computational Intelligence and Security (CIS), 2013 9th International Conference On, 2013, pp. 663-667.

[5] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight internet protocols for web enablement of sensors using constrained gateway devices," in Computing, Networking and Communications (ICNC), 2013 International Conference On, 2013, pp. 334-340.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, pp. 2347-2376, 2015.

[7] U. Hunkeler, H. L. Truong and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in Communication Systems Software and Middleware and Workshops, 2008. Comsware 2008. 3rd International Conference On, 2008, pp. 791-798.

[8] A. Banks and R. Gupta, "MQTT Version 3.1. 1," OASIS Standard, 2014.

[9] "TLS overhead, Available: http://netsekure.org/2010/03/tls-overhead/," MAR 12TH, 2010.

[10] Hive MQ, "MQTT Security Fundamentals: TLS / SSL, Available: http://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl," .

[11] M. Collina, G. E. Corazza and A. Vanelli-Coralli, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST," in 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC), 2012, pp. 36-41.

[12] S. Lee, H. Kim, D. Hong and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," in The International Conference on Information Networking 2013 (ICOIN), 2013, pp. 714-717.

[13] ITU-T, "X.800 Recommendation," 1991.

[14] S. Blake-Wilson, B. Moeller, V. Gupta, C. Hawk and N. Bolyard, "Elliptic curve cryptography (ECC) cipher suites for transport layer security (TLS)," 2006.

[15] J. H. Chung and T. H. Cho, "A Method to Improve Energy Efficiency for IoT Using SSL/TLS on Wireless Network," Journal of the Korea Institute of Information Security and Cryptology, vol. 26, pp. 661-666, 2016.

[16] IoTF Standard, " The Classification and Security Requirements based on IoT Device Capabilities," IoTFS-0081, 12.1, 2015.

[17] T. Dierks, "The transport layer security (TLS) protocol version 1.2," 2008.

[18] G. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic. Prentice hall New Jersey, 1995.

[19] J. Yen and R. Langari, Fuzzy Logic: Intelligence, Control, and Information. Prentice-Hall, Inc., 1998.

[20] R. Babuška, "Fuzzy Systems, Modeling and Identification," Delft University of Technology, Department of Electrical Engineering Control Laboratory, Mekelweg, vol. 4, 1996.

[21] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," Mobile Computing, IEEE Transactions On, vol. 5, pp. 128-143, 2006.
[22] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "Analyzing the energy consumption of security protocols," in Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003, pp. 30-35.
[23] R. Karri and P. Mishra, "Minimizing energy consumption of secure wireless session with QoS constraints," in Communications, 2002. ICC 2002. IEEE International Conference On, 2002, pp. 2053-2057.

Author Profile

Jin Hee Chung received the B.S. degree in computer science from Dankook University in 2015 and now doing M.S. degree in Department of Electrical and Computer Engineering from Sungkyunkwan University.

Tae Ho Cho received a Ph.D. degree in Electrical and Computer Engineering from the University of Arizona, USA, in 1993, and B.S. and M.S. degrees in Electrical Engineering from Sungkyunkwan University, Republic of Korea, and the University of Alabama, USA, respectively. He is currently a Professor in the College of Information and Communication Engineering, Sungkyunkwan University, Korea.