Secure Key Exchange with Authentication using Enhanced Diffie-Hellman and RSA applying to SSL

Junnel E. Avestro, Ariel M. Sison, and Ruji P. Medina

***Abstract*— Protecting data over the internet is a paramount concern. Cryptography plays a significant role in cybersecurity. The Diffie Hellman (DH) and the RSA Algorithms are the basis of several security standards and services on the internet. If the security of both algorithms is compromised, such systems will collapse. In this proposal, the combined cryptography system aims to achieve a secret message exchange. The Diffie-Hellman (DH) algorithm is implemented in securing data on the internet. The DH uses a symmetric key algorithm for shared secret keys between parties over an unsecured channel. The DH vulnerabilities, it does not have an authentication mechanism to validate the key exchange values, including the primitive root, which is essential to secure the communication.**

In this paper, we proposed that Enhanced Diffie-Hellman (EDH) will not generate primitive root (g). Instead, it uses two prime numbers (P and Q) on both parties and performs pre-computation before both parties exchange key values. Another strength of the proposed algorithm, it provides authentication for better security—also, the integration of EDH and RSA algorithm to handle secure encryption and decryption of the data. The proposed algorithm is secure. It encrypts and decrypts the message with secretly generated sender key and receiver key, which is known to the sender and receiver—implemented two-level security.

***Index Terms*— Diffie-Hellman, Public Key Cryptography, RSA, Secure Key Exchange, Authentication**

I. Introduction

THE public-key cryptography, also known as asymmetric encryption, is a form of a cryptosystem that uses both public key and private key, the keys used in encryption and decryption of data. Public key cryptography is composed of digital signatures, key exchange, and data encryption. This cryptosystem helps in achieving data integrity, confidentiality, and authentication. [1] - [3]. Cryptography plays a major role in securing information shared over the internet. In this way, the unauthorized user cannot access the information, and it changes the plain message into a cipher form. [4] - [7].

TABLE I
Work Limitation of Diffie Hellman and RSA

Diffie Hellman	RSA
- not applicable for asymmetric key exchange - not applicable in digital signature - could be used is Denial-of-Service attack - cannot be used to encrypt a message	- prolonged key generation - slow signing and decryption while slightly tricky to implement securely - the key is vulnerable to various attack if poorly implemented

Table I shows the limitation of both DH and RSA algorithms. The authors aim to modify the key-exchange value for both algorithms to increase reliability and security [8], [9].

The DH key exchange is widely used to establish session keys in Internet protocols. It is the primary key exchange mechanism in SSH and IPsec and a popular option in TLS. The DH purpose is to allow two users to securely exchange secret keys for subsequent message encryption [10], [11]. The DH Key exchange algorithm is the public key exchange mechanism of cryptography in which both parties must agree on a secret key sharing unique to the parties communicating data [12] - [16]. The computation of P and Q involves calculating some more common factors, which makes the calculation more complex [17], [18].

Ephemeral DH used in TLS differs from Classic DH (CDH). The CDH key exchange uses the same DH private key. When both parties perform key exchange, they end up with the same shared secret. In Ephemeral DH, it generates key value for every connection and never use the same key. It enables Forward Secrecy (FS) to avoid the server long-term private key get leaked. The Ephemeral DH also aims to replace TLS resource-constrained Internet of Things (IoT) devices using a selection of lightweight ciphers and formats. [19-21].

Elliptical curve cryptography (ECC), with its tenets on the elliptic curve theory, is a general key encryption method for making quicker, shorter, and more effective keys. The elliptic curves vulnerabilities are twist-security attacks and side-channel attacks. These attacks threaten to invalidate the security ECC aims to provide private keys [22] - [24].

Manuscript received November 15, 2019. This work supported by the Graduate Program of the Technological Institute of the Philipines-Quezon City.

Junnel E. Avestro is the student of the Doctor of Information Technology program in the Technological Institute of the Philippines-Quezon City and also currently working as College Professor in the Information Technology department in T.I.P. Quezon City (email: javestro.it@tip.edu.ph).

Ariel M. Sison is a Professor in Doctor of Information Technology program in Technological Institute of the Philippines-Quezon City (email: ariel.sison@eac.edu.ph).

Ruji P. Medina is the Dean of Graduate Program in Technological Institute of the Philippines-Quezon City (email: ruji.medina@tip.edu.ph).

II. Literature Review

A. Diffie Hellman Algorithm

The DH protocol is commonly implemented and deployed that has less security because it has no authentication mechanism. To maintain support for obsolete 1990s-era export-grade crypto, it uses week DH parameters for Servers. [25],[26].The Diffie Hellman Key Exchange (DHKE) protocol was applied in an embedded system and used to analyze the 1024-bit and 2048-bit key timing patterns during the attacks. The implementation of securing Raspberry-pi protocol in embedded systems, reveal several vulnerabilities like lower processing power and high deployment scale. [30], [31],[32].

DH algorithm supports the following:

1. Secure Socket Layer (SSL)
2. Transport Layer Security (TLS)
3. Secure Shell (SSH)
4. Internet Protocol Security (IPSec)
5. Public Key Infrastructure (PKI)
6. Bitcoin and Etherium
7. Online shopping
8. Cellphone communication
9. Used in a card payment system
10. POS ATM network management

B. RSA Algorithm

RSA applied in a network environment cryptography. It supports the following hardware and software:

1. Securing electronic communication and online data storage.
2. Provide a method of assuring confidentiality, integrity, and authenticity of electronic communication.
3. Use to ensure the internet, social media, online shopping & secure personal information.
4. It is used in security protocols like IPSEC/IKE, TLS/SSL, PGP, SSH, and SILC.
5. The government and military used it to secure communication.
6. It is used for signing a digital signature.
7. It is implemented on a website and web-based application.
8. High-speed & simple encryption.
9. It is easier to implement and understand.
10. Widely deployed, better industry support & prevents the third party from intercepting the message [9], [27]-[29].

C. Secure Sockets Layer (SSL)

SSL the gold standard for keeping an internet connection secure and also safeguarding the sensitive data between two systems, preventing the unauthorized user from reading and modifying any information transferred, including potential personal details. It guarantees the transferred data between systems or sites and users, and it remains secure and impossible to read, it uses encryption algorithm is implemented to scramble data during transmission, deter the unauthorized user from acquiring it [33].

III. Proposed Algorithm

The proposed integration of the EDH and RSA algorithm is composed of two parts. The EDH algorithm generates key exchange value while the RSA algorithm encrypts and decrypts the message. The combination of EDH and RSA algorithm will be integrated into SSL protocol.

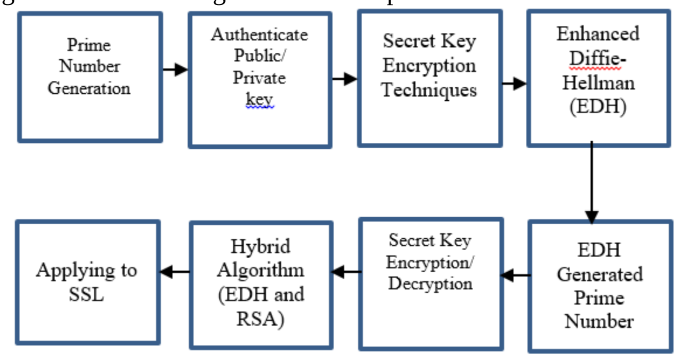

Fig. 1. Flowchart Diagram of EDH with RSA Applying to SSL.

First part:

A. Secret key generation

Secret key generation is using the EDH algorithm. These are the steps:

1. User A generates a random prime number (P), and then it will be multiplied by 2 (Pn). Therefore Pn = P * 2 (to avoid using primitive root, the value of P will be twice) and sends it to User B.
2. The User B receives Pn and calculate it as P = Pn/2 (to return the user A original prime number (P)). After this, User B also generates a random prime number (Q). User, B makes Q as Qn = P + Q (to add P to Q) and send it back to User A.
3. User A received the number from User B and subtracted P to Qn.
   User A get Q = Qn-P and assign this value of Qn to Q

B. Authentication process

4. User B Authenticates and Send Public key to user A and receive the value of Qn.
   Q’ = Qn-P and then compare with the value of Q
   If Q'==Q are equal, it will continue. Otherwise, it will stop to prevent compromised conversation.
   If the value of Q matches then User B generate this public key (PubB)
   Select Random Private Prime Number Pb.
   PubB = P^{Pb} mod Q
   PubB is sent to User A
5. User A receives Public Key of User B, and then User A also generates a public key (PubA).
   Select Random Private Prime Number Pa
   PubA = P^{Pa} mod Q
   PubA sent to User B
6. User B receives the Public Key of User A and Secret Key Generation process
   SecKb = $PubB^{Pa}$ mod Q (1)
7. User A Secret Key Generation Process
   SecKa = $PubA^{Pb}$ mod Q (2)

Second Part: Sender key and Receiver Key Generation

The sender key will be used for message encryption while the Receiver key will use for decryption to the receiver message.

8. Calculate the new Prime number (Pnew)
   For User A

Pnew = P x SecKa (the value P originated from step 1, and SecKa is from (2)
Find next prime of Pnew and which new value of P of User A
P = next prime(Pnew) (3.1)
For User B
Pnew = P x SecKb (SecKb originated from (1)
Find next prime of Pnew and which new value of P of User B
P = next prime(Pnew) (3.2)
Notice that (3.1) and (3.2) are the same value because the secret key must identical in both User A and User B

9. Calculate new Q
For User A
Qnew = Q x SecKa (the value of Q originated from step 2, and SecKa is from (2)
Find the next prime of Qnew, which the new value of Q at User A
Q = nextPrime(Qnew) (4.1)
For User B
Qnew = Q x SecKb (SecKb originated from (1))
Find the next prime of Qnew, which the new value of Q at User B
Q = nextPrime(Qnew) (4.2)

The RSA Algorithm
The values of Prime number P and Q respectively originated from EDH to avoid RSA to generate random prime numbers to increase security.

10. Calculate the value of N and φN
$N = P \times Q$ (5)
$\varphi N = (P-1)\times(Q-1)$ (6)
11. Choose e such that $1 < e < \varphi N$ and e and N are co-prime
12. Compute the value for d such that $d \times e \bmod \varphi N = 1$
13. Sender key is (e, N); Receiver key is (d, N)
14. The encryption of message (m) is
$c = m^e \bmod N$ (7)
15. The decryption of message (c) is
$m = c^d \bmod N$ (8)

IV. SIMULATION AND RESULT

The classic Diffie-Hellman algorithm compared to the modified Diffie-Hellman based on the following criteria: using 64-bits, 128-bits, 1024-bits, and 2048-bits symmetric keys lengths.

TABLE II.
COMPARISON OF CDH AND EDH

Symmetric key	**Enhanced DH**	Classic DH	Time Difference %
64 – bits	**27.72 sec**	88.58 sec	68.70 %
128– bits	**66.71 sec**	165.55 sec	59.70 %
512– bits	**271.96 sec**	603.87 sec	54.96 %
1024– bits	**1722.82 sec**	2450.66 sec	29.70 %
2048– bits	**18551.22 sec**	23352.92 sec	20.56 %

Table II shows that Enhanced Diffie-Hellman is much faster amongst symmetric keys versus Classic DH.

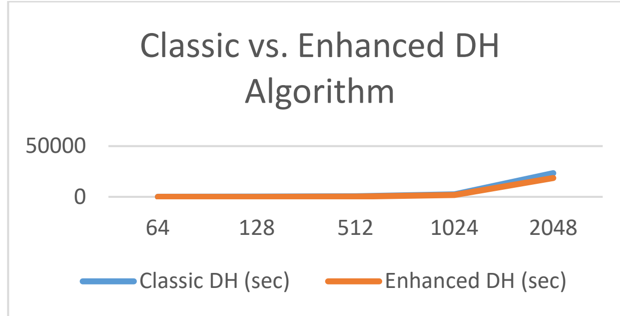

Fig. 2. Classic and Modified Diffie-Hellman Comparison.

Fig. 2 shows the performance comparison of the CDH and the EDH algorithms in terms of run time. The EDH is faster in all symmetric key lengths, ranging from 64-bits up to 2048 bits since the EDH does not require to generate primitive root. The time needed is significantly shorter compare with CDH in different symmetric key lengths. Since RSA generate two big prime number, the key generation is slower. The proposed algorithm will use the two prime numbers generated by EDH to speed up the keys needed by RSA. In that way, it gets faster key generation, encryption, and decryption of messages—the throughput in the actual scenario improved. More transactions can accommodate.

A. Program Simulation

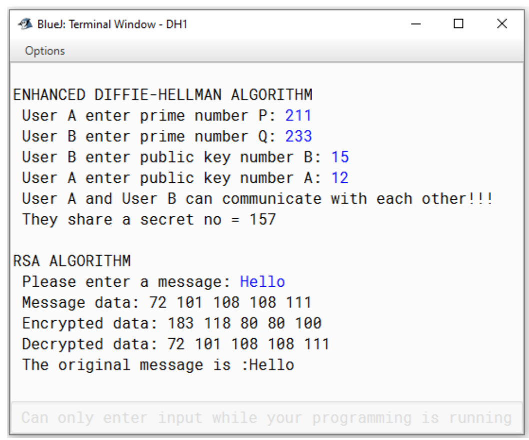

Fig. 3. Integration of EDH and RSA algorithm.

Fig. 3 shows the EDH and RSA algorithms. The first part is EDH. It allows the user to input two prime numbers, public keys, and generates a secret key. The second part is that the RSA algorithm uses the two prime numbers (P and Q) generated by EDH. The program allows the user to compose message data, encrypt, decrypt, and return the original message.

B. Time Complexity

The CDH and EDH time complexity comparison depicts below.

The CDH will undergo the following process:

Step 1: Choose prime numbers (P) and primitive root (g)

Step 2: User A selects a secret no(a) and computes g^a mod p, let us call it A. User A sends A to User B.

Step 3: User B selects a secret no(b) and computes g^b mod p, let us call it B. User B sends B to User A.

Step 4: User A computes S_A = Ba mod p

Step 5: User B computes S_B = Ab mod p

Step 6: If S_A=S_B, then User A and User B can agree for future communication.

It has constant values for the following:

A prime number (P) → O (1)

Primitive root (g) → O (n)

User A Secret number (a) → O (1)

User B Secret number (b) → O (1)

User A Secret key (A) → O (1)

User B Secret key (B) → O (1)

User A Shared key (S_A) → O (1)

User B Shared key (S_B) → O (1)

The CDH has seven constants value and one variable, which the primitive root generation (g). It will come with a big O (7+n).

The Enhanced DH undergo the following process:

Step 1: Generation of two prime numbers (P, Q). Perform computation and validation
Step 2: User A selects a secret number(A) and computes PubA= P^{Pa} mod Q, let us call it PubA. User A sends PubA to User B.
Step 3: User B selects a secret number(B) and computes PubB=P^{Pb} mod Q, let us call it PubB. User B sends PubB to User A.
Step 4: User Al computes SecKa = $PubA^{Pb}$ mod Q
Step 5: User B computes SecKb = $PubB^{Pa}$ mod Q
Step 6: If SecKa == SecKb S_B, then User A and User B can agree for future communication.

It has constant values for the following:
Prime number A (P) → O (1)
Computed User A Prime (Pn) → O (1)
Primitive root B (Q) → O (1)
Computed User A Prime (Qn) → O (1)
Validate Q ‘ == Q → O (1)
User A Secret number (A) → O (1)
User B Secret number (B) → O (1)
User A Public key (PubA) → O (1)
User B Public key (PubB) → O (1)
User A Shared key (SecKa) → O (1)
User B Shared key (SecKb) → O (1)

The EDH has eleven constants. It comes up with a big O (1). The CDH algorithm took four seconds, and the EDH algorithm took three seconds for execution.

C. SSL Testing

The time cost-related before the exchange of data via TLS, both parties must agree
on the connection parameters: the version of the protocol used, the method of data encryption, and also to check the certificates, if necessary.

TABLE III
WORK TEST OF SSL PROTOCOL RESULT

Protocol	Cipher	Hash	Key exchange	Request Per second
SSL2	RC4 40-bits	MD5	RSA 512 bits	79
SSL3	RC4 56-bits	SHA1	RSA 512 bits	85
TLS	RC4 40-bits	SHA1	RSA 512 bits	92

Table II shows that when the protocol version and the encryption method are approved, the client checks the submitted certificate and initiates the exchange of keys based on RSA or Diffie Hellman, depending on the set parameters. Simultaneously, the performance of the software implementations of the RSA algorithm is low and quickly decreases with increasing key length.

V. CONCLUSION

The proposed algorithm is a secure and detailed process of a cryptosystem. It is easy to understand and more reliable as the algorithm complexity is a logarithmic problem to solve. Security of RSA depends on the prime factorization of N. The proposed algorithm authenticates a user as shown is the first part of the algorithm by exchange value of random P and Q. Then generate secret key at both sides. Secret keys are not known or sent over a communication channel, so eavesdropper has no chance to get the value of the secret key. This value of P and Q secretly generated at both sides, and this value is not shared through the communication channel, so we claim that the proposed method is more secret than the original RSA. The disadvantage of the proposed algorithm has extra steps that mix symmetric key and asymmetric key cryptography. The time required for this integration steps is disadvantageous of this proposed method. But we can achieve better security

Also, the SSL protocol is used to secure communication over the internet. However, the protocol has several weaknesses and limitations, which leads to various vulnerabilities. The proposed algorithm is secure, two-level of security implemented. The algorithm is based on hybrid cryptography, it uses asymmetric in the sender, and receiver key and symmetric key that is both users A and user B uses the same key pair for encryption and decryption of the message.

REFERENCES

[1] L. K. Galla, V. S. Koganti and N. Nuthalapati, "Implementation of RSA," *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kumaracoil, 2016, pp. 81-87.

[2] B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, and D. Jao, "Post-Quantum Cryptography on FPGA Based on Isogenies on Elliptic

Curves," in *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 64, no. 1, pp. 86-99, Jan. 2017.

[3] M. Lakkadwala and S. Valiveti, "Parallel generation of RSA keys — A review," *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence,* Noida, 2017, pp. 350-355.

[4] Karthik, Chinnasamy, and Deepalakshmi, "Hybrid cryptographic technique using OTP: RSA," *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS),* Srivilliputhur, 2017, pp. 1-4.

[5] B. J. S. Kumar, V. K. R. Raj, and A. Nair, "Comparative study on AES and RSA algorithm for medical images," *2017 International Conference on Communication and Signal Processing (ICCSP),* Chennai, 2017, pp. 0501-0504.

[6] S. Mathur, D. Gupta, V. Goar, and M. Kuri, "Analysis and design of enhanced RSA algorithm to improve the security," *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT),* Ghaziabad, 2017, pp. 1-5.

[7] I. G. Amalarethinam and H. M. Leena, "Enhanced RSA Algorithm with Varying Key Sizes for Data Security in Cloud," *2017 World Congress on Computing and Communication Technologies (WCCCT),* Tiruchirappalli, 2017, pp. 172-175.

[8] E. E. Classes, "Diffie Hellman Key Exchange in Hindi for Symmetric Key Encryption System –With Example," 2016. [Online]. Available: https://www.youtube.com/watch?v=_M2Ea_3DRGA. [Accessed 24 03 2017].

[9] K.Suganya, "Performance study on Diffie Hellman," *INTERNATIONAL JOURNAL FOR RESEARCH IN AP PL I ED SC IENC E*, vol. 2, no. 3, pp. 68-75, 2014.

[10] Nan Li, "Research on Diffie-Hellman key exchange protocol," *2010 2nd International Conference on Computer Engineering and Technology,* Chengdu, 2010, pp. V4-634-V4-637.

[11] P. Deshpande, S. Santhanalakshmi, P. Lakshmi, and A. Vishwa, "Experimental study of Diffie-Hellman key exchange algorithm on embedded devices," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS),* Chennai, 2017, pp. 2042-2047.

[12] Yusfrizal Yusfrizal et al., .2018. Key Management Using Combination of Diffie–Hellman Key Exchange with AES Encryption. *The 6th International Conference on Cyber and IT Service Management (CITSM 2018)*

[13] A. Taparia, S. K. Panigrahy, and S. K. Jena, "Secure key exchange using enhanced Diffie-Hellman protocol based on string comparison," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),* Chennai, 2017, pp. 722-726.

[14] P. Joshi, M. Verma, and P. R. Verma, "Secure authentication approach using Diffie-Hellman key exchange algorithm for WSN," *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* Kumaracoil, 2015, pp. 527-532.

[15] A. Jalali, R. Azarderakhsh, M. M. Kermani, and D. Jao, "Supersingular Isogeny Diffie-Hellman Key Exchange on 64-Bit ARM," in *IEEE Transactions on Dependable and Secure Computing,* vol. 16, no. 5, pp. 902-912, 1 Sept.-Oct. 2019.

[16] M. N. Mejri, N. Achir, and M. Hamdi, "A new group Diffie-Hellman key generation proposal for secure VANET communications," *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC),* Las Vegas, NV, 2016, pp. 992-995.

[17] P. K. Panda and S. Chattopadhyay, "A hybrid security algorithm for RSA cryptosystem," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS),* Coimbatore, 2017, pp. 1-6.

[18] A. Karakra and A. Alsadeh, "A-RSA: Augmented RSA," *2016 SAI Computing Conference (SAI),* London, 2016, pp. 1016-1023.

[19] A. Bruni et al. *Formal Verification of Ephemeral Diffie-Hellman Over COSE (EDHOC)*. 2018

[20] P. M. Aiswarya, A. Raj, D. John, L. Martin, and G. Sreenu, "Binary RSA encryption algorithm," *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* Kumaracoil, 2016, pp. 178-181.

[21] Why use Ephemeral Diffie-Hellman? 2018 [Online]. Available: https://tls.mbed.org/cryptography/ephemeral-diffie-hellman

[22] C. Varma, "A Study of the ECC, RSA and the Diffie-Hellman Algorithms in Network Security," *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT),* Coimbatore, 2018, pp. 1-4.

[23] N. Mehibel and M'hamed Hamadouche, "A New Approach of Elliptic Curve Diffie-Hellman Key Exchange," *The 5th International Conference on Electrical Engineering* – Boumerdes (ICEE-B) October 29-31, 2017, Boumerdes, Algeria

[24] A. Karakra and A. Alsadeh, "A-RSA: Augmented RSA," *2016 SAI Computing Conference (SAI),* London, 2016, pp. 1016-1023.

[25] David Adrian et al., Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. *Communications of the ACM,* January 2019, Vol. 62 No. 1, Pages 106-114

[26] H. Bodur and R. Kara, "Implementing Diffie-Hellman key exchange method on logical key hierarchy for secure broadcast transmission," *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN),* Girne, 2017, pp. 144-147.

[27] F. H. a. F. R. Michael Cobb, "RSA algorithm (Rivest Shamir Adleman)," 2014.[Online]. Available:http://searchsecurity.techtarget.com/definition/RSA. [Accessed 24 03 2017]

[28] H. M. J. Ali Makhmali, "Comparative Study On Encryption Algorithms," *International Journal of Scientific & Technology Research,* vol. 2, no.6, p. 44, 2013.

[29] D. Chauhan, "RSA and Diffie Hellman Algorithm," 2016 [Online].Available:https://www.slideshare.net/daxeshchauhan/rsa and Diffie hellman algorithms 64170629. [Accessed 24 03 2017].

[30] Y. F. Alias and H. Hashim, "Timing analysis for Diffie Hellman Key Exchange In U-BOOT using Raspberry pi," *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),* Penang, 2018, pp. 212-216.

[31] Poltak Sihombing, Jos Timanta Tarigan, Benyamin Ginting, and Dahlan Sitompul. Security System Based on Vibration and Infra-Red Sensors Using Raspberry *INTERNETWORKING INDONESIA JOURNAL 2019.*

[32] Daisuke Yamauchi, Yoshihiro Ito, and Toru Tahara. A Method of QoE Management in Online Shopping Web Services with TCP Variables. *INTERNETWORKING INDONESIA JOURNAL 2015.*

[33] https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https

Junnel E. Avestro is currently taking up Doctor in Information Technology at the Technological Institute of the Philippines Quezon City. He earned his Master in Information Technology degree in 2005 and obtained the Bachelor of Science in Computer Engineering at Technological Institute of the Philippines Quezon City in 1997. He is a professor in Information Technology in the Technological Institute of the Philippines Quezon City.

He has almost 20 years of teaching experience in the area of Computer Engineering and Information Technology. His research interest includes cybersecurity, microprocessor system, embedded system, and mobile application development. He has numerous publications in various fields and translated them into a mobile application.

Ariel M. Sison graduated with higher honors in the Technological Institute of the Philippines Quezon City in 2013 with the degree of Doctor of Information Technology. In 2006 at De La Salle University-Manila, he earned his Master of Science in Computer Science. In 1994 in Emilio Aguinaldo College Manila, he obtained his degree in Bachelor of Science in Computer Science. His research interests include data security and data mining. His affiliations are the Philippine Society of IT Educators and Computing Society of the Philippines, International Association of Engineers. He is also a Technical Committee Member of International Academy, Research, and Industry Association for the International Conference on Systems.

Ruji P. Medina He is currently the Technological Institute of the Philippines in Quezon City Graduate Program Dean and earned his Doctoral degree in Environmental Engineering at the University of the Philippines. He worked on the synthesis of nanocomposite material at the University of Houston, Texas.