An Enhanced Mutual Authentication Scheme Based on ECDH for IoT Devices Using ESP8266

Anothay Phimphinith
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
e-mail: mee_gamma@hotmail.com

Xiong Anping, Qingyi Zhu, Yi Jiang, Yong Shen
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
e-mail: xiongap@cqupt.edu.cn

***Abstract*—As a revolutionary and profound technology, the Internet of Things (IoT) has the potential to fundamentally transform our society by simply connecting sensors and smart devices to the Internet. It is no doubt that the security of communications between smart devices is an important issue in IoT. In this paper, we deal with the security scheme for communications between ESP8266 modules, which can provide embedded Wi-Fi capabilities at a low cost. Based on an existed security scheme for ESP8266, we proposed an enhanced mutual authentication mechanism and ECDH-key agreement on curve25519. Compared with the existed schemes, security analysis and performance evaluation show that the new scheme can resist various communication attacks, saying modification attacks, replay attacks, and man-in-the-middle attacks.**

Keywords-ECDH; mutual authentication; curve25519; IoT; ESP8266

I. INTRODUCTION

The Internet of Things technology is leading a better life by using the physical devices, and other appliances embedded with software, sensors, and connectivity which enables these things to connect and exchange data. Many different protocols provided the devices connectivity, such as Message Queuing Telemetry Transport (MQTT), are designed for remote location connections with limited network bandwidth. The preventive security mechanisms are used to secure IoT communication, such as device identity management, encryption, and access control as well as device auditing and monitoring. This technology can be applied in different fields, ranging from industry, infrastructure, and agriculture to many other smart applications. Due to the ubiquitous connectivity, the security of communication between IoT devices plays a more and more important role in IoT.

The ESP8266 module is a system on a chip (SoC) by means of the device itself contained microcontroller capability and full TCP/IP stack which allowed for single-chip devices capable of connecting to Wi-Fi. ESP8266 is used to provide the IoT systems with embedded Wi-Fi capabilities at the lowest cost with the greatest functionality [1]. Since the devices are interrelated and uncovered to many vulnerable attacks, such as eavesdropping, tampering, and jamming attack [2], hence more security measures are needed. Recently many techniques, such as the symmetric key encryption [3], and the asymmetric cryptography [4], [5], [6], have been used to enhance the security of IoT.

The purpose of this paper is to propose an improve security model using Elliptic-curve Diffie–Hellman (ECDH) suitable for ESP8266 based on Kodali and Naikoti 's model [7]. In their model, the NIST P-192 curve is used for ECDH key exchange which is suitable to the low-cost device environment. However, the curve P-192 is disapproved for key establishment, according to NIST-SP 800-56A recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography [8]. While selecting the safe curves for elliptic-curve cryptography [9], [10], the following criteria should be considered: efficiency, security and lightweight.

In the public-key cryptography, the strength of the cryptography method depends on the difficulty of solving the mathematical problems. The security of elliptic-curve cryptography (ECC) depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points. The difficulty of the problem is determined by the size of the elliptic curve [11].

To achieve the authentication and authorization, using ECC seems to be the best choice, due to the limited storage capacity of ESP8266. The complementary which makes this module popular are inexpensive, more compatible development environments, flexible design, and enhanced function. According to the pros, ESP8266 families are popular for many IoT applications, such as flooding detection system, data center temperature monitoring, home automation system. Recently, ESP8266 is also introduced in industry 4.0 based on service quality and transmission reliability [12]. The module of ESP8266 ranking from ESP8266-01 to 14 which improves over the previous one with reference to the hardware of the module. The various features include the number of GPIO pins, the types of pins, antenna on the chip and presence of shield, types of the modules packaging and storage capacity. The ESP8266 module requires sufficient power supply with 3.3V – 3.6V and >= 250 mA [12]. It is power efficient and low in cost.

The rest of this paper is organized as follows. Section II presents the related work. In Section III, review the preliminaries of related knowledge. The design of our scheme and its implementation are presented in Section IV.

978-1-7281-2184-0/19/$31.00 ©2019 IEEE

The security analysis and performance evaluation are shown in Section V. Finally, conclusions are given in Section VI

II. Related Work

To guarantee the broadcast over an insecure network, the authentication and key agreement protocol have been widely used in IoT. Arasteh et al. [13] introduced a new lightweight authentication protocol for WSNs using session key agreement and new sensor node registration in 2016, which has become the foundation of the follow-up work. Later, Fan and Niu [14] pointed out that the scheme in [13] has some security weaknesses, for example, it cannot prevent the malicious attack. Hence they proposed a new scheme with the fundamental agreement which is secure and robust against the malicious attack. In another study, Jiang et al. [15] proposed an improved scheme over the security weaknesses of Amin et al.'s authentication scheme for WSNs [16] which is vulnerable to offline guessing attack and tracking attack. However, the computation cost of the new scheme [15] is comparatively high as a result of using the Rabin cryptosystem.

To solve the eavesdropping problem in communications between the constrained devices, many scholars have done a lot of work. In 2016, Goyal and Sahula [17] presented a suitable lightweight security algorithm for low power IoT devices. In [12], different public-key cryptosystems, including ECDH, RSA, and ECC for IoT gadgets, are compared, and analysis shows that ECDH has better performance than other algorithms. ECC has been standardized by many international organizations [8], [18], [19], [20]. In 2015, Seo et al. [4] proposed a light-weight authentication method of Transport Layer Security (TLS) handshake using ECDH for local Session Initiation Protocol (SIP) environment. This scheme can improve the overhead occurring at SIP call set-up time. In 2017, Hammi et al. [3] designed a security protocol for WSNs based on symmetric encryption algorithm (Advanced Encryption Standard Galois/Counter Mode).

Reference [5] proposed an anonymous ID-based user authentication with key agreement on ECC for smart cards. However, the mutual authentication cannot be achieved in their protocol. Later, Zhang et al. [21] proposed a new version of anonymous authentication with key agreement protocol used for client-server environment to address the flaws of the protocol in [5]. The protocol in [21] is efficient and can provide more features than the protocol in reference [5]. Teguig et al. [22] introduced a new mechanism for public keys management using an elliptic curves cryptosystem which provides 161 bits security. In 2018, Li et al. [6] proposed a symmetric cryptography and hash based user authentication protocol with privacy protection for WSN in industrial Internet of Things (IIoT) environment. In this paper elliptic curve cryptography (ECC) is introduced to the design of authentication for IoT, since it is more efficient, and with much shorter key length than RSA while achieving the same security level.

Recently, Kodali and Naikoti [7] introduced a security model based on public-key exchange by using ECDH on NIST P-192 curve for IoT using low-cost devices. Their model provides several security and functional features with high efficiency. However, this scheme is unable to resist replay attack, since the curve NIST P-192 is unsecured, which means that attackers can intercept the data and re-transmit it to users successfully. In this paper, the author would like to demonstrate how to mount replay attacks based on the previous scheme. The main contributions of this paper are as follows:

1. To overcome the flaws of the security model in [7], present a curve25519 based mutual authentication between smart devices with ECDH.
2. The design of a new security model with two-way authentication by trusted authority verifying the risky node during exchange key between two devices.
3. The comparisons of security properties and performance with related authentication and key exchange protocols show that, our scheme is more suitable to resist usual attacks with acceptable computational efficiency.

III. Design of Our Scheme

In this section is a review Kodali and Naikoti's secutity model and an introduction a new security model with two-way authentication mechanism by trusted authority verifying the risky node during exchange key between two devices And Elliptic curve Diffie-Hellman-based key exchange is used for securing communications between ESP8266 modules.

A. Review Kodali and Naikoti's Security Model

Their security model is based on Elliptic Curve Diffie-Hellman (ECDH) on NIST P-192 curve [7]. The element of each parameter is shown in figure 1.

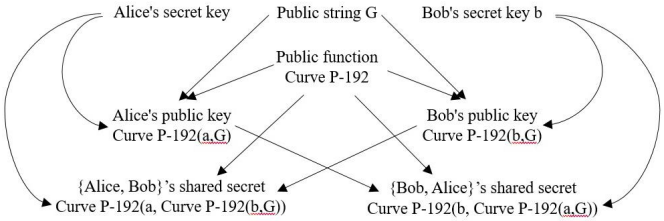

Figure 1. Kodali and Naikoti's scheme.

The following is an explanation of public key exchange procedures shown in Figure 1.

- At first, two parties choose the public function curve p-192 over a prime field F_p, and public string G.
- Alice selects an integer a randomly, then performs point multiplication with G and calculates the public key $T_a = curve\ P\text{-}192(a, G)$ and sends T_a to Bob.
- Bob also selects an integer b randomly, then performs point multiplication with G and calculates the public key $T_b = curve\ P\text{-}192(b, G)$, and sends T_b Alice.
- Alice calculates:
  $aT_b = curve\ P\text{-}192(a, curve\ P\text{-}192(b, G))$
- Bob calculates:
  $bT_a = curve\ P\text{-}192(b, curve\ P\text{-}192(a, G))$

B. Our Scheme

In our security model, the trusted authority (TA) as a third-party is introduced to support monitoring during exchange key between two devices. Every device of our IoT network should be registered in the TA firstly, and then TA will generate an authentication key with the identity of the device and a random token. This allows the devices to verify each other's during exchange keys that correspond to the certified devices. As shown in Figure 2, the IoT network architecture of our model consists of three layers: the perception layer, the network layer and the application layer. The trusted authority is in application layer. The devices are located in perception layer, and can communicate with others through network layer.

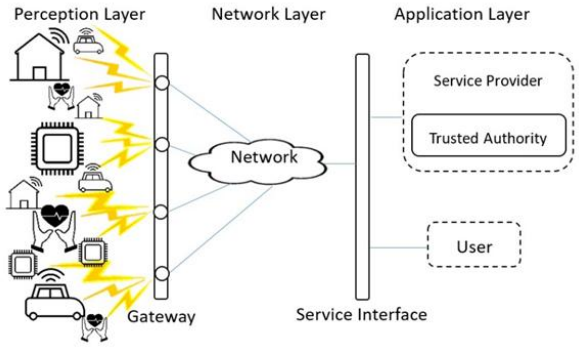

Figure 2. IoT network architecture.

The new model is based on ECDH key agreement protocol and two-way authentication scheme, which consists of two sections: the registration phase, and the mutual authentication and key agreement phase. Table I outlines the notations used in the remainder of this paper, and the detailed description of the proposed scheme are given in the following subsections.

TABLE I. NOTATIONS USED IN THIS PAPER

Symbol	Description
a, b	random digital number
G	generator point
Y_a, Y_b	public key parameter
ID_i	the unique identifier ID of device i
S_i	the key of device i for authentication
S_i`	the authentication key of device i for key exchange
$h()$	the one-way hash function
$token_i$	the unique random number of each key authentication
T_i	the time stamp of device i
S_{Ti}	Trusted Authority session authentication key for device i

1) Registration phase

In the registration phase, we proposed a key management based on "personal identity". The principle of this method is detailed in Figure 3. The devices register on Trusted Authority (TA) with a unique identifier *ID*, and TA generates $token_i$ for each registered device to create a secret key for the authentication of S_i. S_i is obtained from ID_i and $token_i$ using the "PersoFunc ()" function (see equation 1). It is an irreversible function that generates a strong key, and protects the $token_i$ against deductive attacks.

$$S_i = PersoFunc\ (token_i \oplus ID_i) = h\ (tokeni \oplus ID_i) \quad (1)$$

Once S_i is created and set into the device, the device is able to be associated with the ESP8266 network.

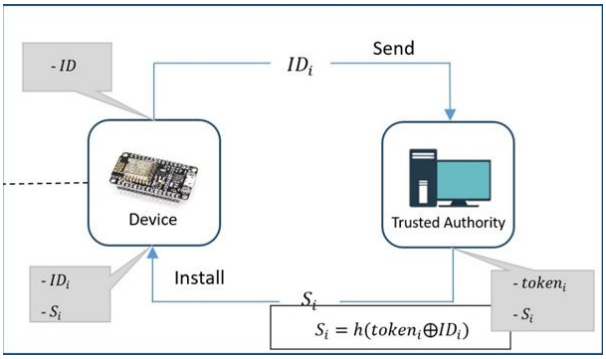

Figure 3. the personal identity of devices.

The personal identity is aimed to ensure the communications between a device A and TA cannot be interrupted by any other devices in the same ESP8266 network. Thus each device has a secret identity in our IoT system. In addition, this personal identity function has more advantage. Even if an attacker could get an authentication key EF of one device, it will not influence the security of the rest of devices belonging to the same system network. Because of each device has unique identifier which provide by TA.

2) Mutual authentication and key agreement phase

In this section is an adoptable the authentication approach presented in [4], [3], [14], [23] to ensure security under the communication attacks. Let S_i denotes the authentication key agreement which will install in concerned devices and Trusted Authority (TA). The ECDH based key exchange on cuvre25519 for devices.

As shown in Figure 4, the 4-way handshake process of authentication and key exchange between devices and TA is summarized as following.

Step 1: The device A computes the authentication key S_A` from $S_A$ by using hash function, where $S_A$ derives from TA. The public key $Y_a$ is obtained by ECDH, where $Y_a = aG$. Then A sends the generated keys $S_A$`, Y_a, ID_A to B as the authentication response.

Step 2: B gets an association request ($ID_A||S_A`||Y_a$) from A, then separates it for next process. The device B requests the key S_{TA} from TA, by sending the ($ID_A||S_B$) to check whether A belongs to the system.

Step 3: TA returns the request to B if the device is not blacklisted, that is to say, TA will check $S_B^* ?= S_B$ whether ID_B is registered in the system or not. If it is registered, TA will extend the process of authentication key management. Then TA will create the key S_{TA} (see equation 4) by hashing

the combine value of keys $S_A`$ and time stamp $T_A$, where $S_A`$ is hash of S_A (see equation 3). S_{TA} will be used for the association request of the exchange key encryption in the unicast mode. Finally, the trusted authority will transfer $(S_{TA}||T_A)$ to device B.

$$S_A = h\ (token_i \oplus ID_A) \quad (2)$$

$$S_A` = h\ (S_A) \quad (3)$$

$$S_{TA} = h\ (S_A` \oplus T_A) \quad (4)$$

Step 4: After receiving the $(S_{TA}||T_A)$ from TA, B will compare the S_{TA} with $h\ (S_A` \oplus T_A)$ (see equation 5). If $S_{TA}$ is equal $h\ (S_A` \oplus T_A)$, it means that A is an authenticated device registered in the system. Then B will accept the public key Y_a from A and send his public key $Y_b||ID_B$ to A. Otherwise, the association request will be stopped.

$$h\ (S_A` \oplus T_A) ?= S_{TA} \quad (5)$$

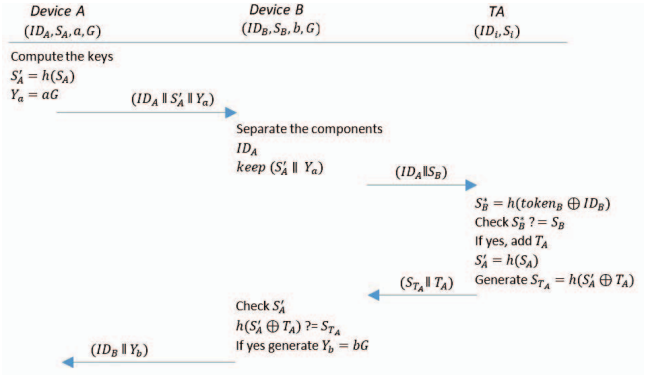

Figure 4. The key exchange protocol.

In the end of the key exchange operation, the device A performs point multiplication between private key of A and public key of B, and computes $aY_b = abG$. Meanwhile, the device B computes $bY_a = baG$ then both of them have the same encrypt and decrypt key which enable them to communicate securely.

Remark 1: Enhance security of our scheme.

To ensure our scheme cover the full security, the enhance 6-way handshake via the upper scheme is not respond the sufficient authentication on both side. By means of device A and B are not sure one another is risky or not, so this problem will be covered in this next process. The full security based on sufficient authentication is describe in (Figure 5). We can summarize this process as following:

The first three steps are as the same as upper scheme show in (Figure 4).

In Step 4: B receives the authentication key $(S_{TA}||T_A)$ from TA which needs to checkup A by checking the key $S_A`$ from A. Then, B compares the $S_{TA}$ with $h\ (S_A` \oplus T_A)$ (see equation 4). Later, device B need to calculate his own authentication key $S_B`$, where $S_B`$ is computed from S_B by using hash function. After checking A, the device B sends an association request to device A. The request contains the unique ID of B, the authentication key $S_B`$, and the public key Y_b, where $Y_b = bG$ obtained by ECDH.

Step 5: A gets the association request $(ID_B||S_B`||Y_b)$ from B, then separates it for next process. Device A requests the key S_{TB} from TA by sending the $(ID_B||S_A)$ to TA to check whether B belongs to the system or not.

Step 6: TA will return the request to A if the device is not blacklisted, that is to say, TA will check whether ID_A is registered in system or not. If it is registered, TA will create the key S_{TB} (see equation 6) by computing the key $S_B`$, and $T_A$, where $S_B$ is used to create the key $S_B`$ (see equation 7). S_{TB} will be used for the association request of the exchange key encryption in the unicast mode.

$$S_B` = h(S_B) \quad (6)$$

$$S_{TB} = h(S_B` \oplus T_A) \quad (7)$$

Trusted authority will transfer $(S_{TB}||T_A)$ to device A.

Finally, A receives $(S_{TB}||T_A)$ which is used to verify B by checking $S_B`$. Then A compares $S_{TB}$ with $h\ (S_B` \oplus T_A)$. If they match, A will accept Y_b and ID_B. Otherwise, if the retrieved $S_B`$ or $T_A$ or both of them are wrong, which means that the keys have been modified during their transmission, then $h\ (S_B` \oplus T_A)$ and S_{TB} will not match. In this case, the key $S_B`$ will not be accepted, which means that B could not be authenticated, and the association request will be stopped. A denies the pubic key Y_b and ID_B.

In the end of the keys exchange operation we confidence that device A and device B are not risky for system on the point of both device A and device B are honest and be able to communicate to each other securely with the same key $bY_a = aY_b = abG$.

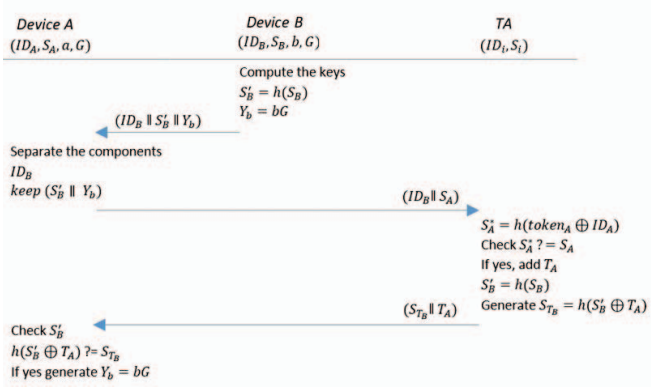

Figure 5. The enhanced key exchange protocol.

IV. Security Analysis and Performance Evaluation

A. Security Analysis

In this section, the proposed algorithm will be inspected for practical negative use cases such as replay attack. It is

observed that in such case the model is able to detect the attacker.

Proposition 1 Registration phase

Proof 1: This proposition is proved through showing that it is impossible for attacker to construct the unique value of S_i. Firstly it shows that the unique value of $token_i$ cannot be constructed by an adversary. Based on the explanation of S_i in figure 4, it can be seen that only the TA can set the value of $token_i$. Then, we show that the unique value $token_i$ has to be a component in the expression of S_i. The expressions of S_i can be computed as follows.

$$S_i = h\ (token_i \oplus ID_i)$$

Thus, an adversary cannot calculate the value of S_i which means that he cannot know the value of $token_i$. Hence, the attacker cannot get S_i through ID_i by mean of a replay attack.

Proposition 2 Mutual authentication and Key agreement phase are analyze the whole attack process to prove that our scheme is safe and robust, and it can resist the Man-in-The-Middle attack.

Proof 2:

Case 1. The attacker poses as device A.

- Step 1: The attacker can get ID_A and generate new values $(S_X`||Y_x)$ for key exchange and send $(ID_A||S_X`||Y_x)$ to device B.
- Step 2: Device B need to check it by sending the request $(ID_A||S_B)$ to TA.
- Step 3: TA sends the response $(S_{TA}||T_A)$ to the request device.
- Step 4: Final verification could be done by device B comparing the two keys. If they do not match, it means that the key $S_X`$ is wrong, S_{TA} is not pairing and the TA is not authenticated, then the association operation will be stopped.

$$h\ (S_X` \oplus T_A)? = S_{TA}$$
$$h\ (S_X` \oplus T_A) \neq h\ (S_A` \oplus T_A)$$

Case 2. The attacker poses as device B.

- Step 1: Device A sends $(ID_A||S_A`||Y_a)$ to the attacker.
- Step 2: the attack needs to check it by sending the request $(ID_A||S_A)$ to TA.
- Step 3: TA sends the response $(S_{TA}||T_A)$ to request device.
- Step 4: The attacker can compares the two keys. If they match, it means that $S_A`$ is correct, thus S_{TA} is pairing and the TA is authenticated.
  At the same time attacker can get ID_B and generate new values $(S_Y`||Y_y)$ for key exchange and send $(ID_B||S_Y`||Y_y)$ to device A
- Step 5: device A needs to check it by sending the request $(ID_B||S_A)$ to TA.
- Step 6: TA sends the response $(S_{TB}||T_B)$ to request device.

Final verification could be done by device A comparing the two keys. If they do not match, which means that $S_Y`$ is wrong, S_{TB} is not pairing and the TA is not authenticated, then the association operation will be stopped.

$$h\ (S_Y` \oplus T_A)? = S_{TB}$$
$$h\ (S_Y` \oplus T_A) \neq h\ (S_B` \oplus T_A)$$

At the end of key exchange process, it shows that device B (attacker) does not belong to system. Thus, device A needs to update the keys to make sure the attacker cannot use $(ID_A||S_A`)$ for next broadcast, since the attacker obtains $(ID_A||S_A`)$ from A. The key can be updated through re-registration to get new unique value of S_A, new $token_A$.

TABLE II. COMPARISON OF SECURITY FEATURES AND FUNCTIONALITY

	Schemes				
	Kodali [7]	*Li et al.* [6]	*Qi* [24]	*Park* [23]	*Our*
Mutual authentication	No	Yes	Yes	Yes	Yes
Key agreement	Yes	Yes	Yes	Yes	Yes
Inside attack resistance	No	No	No	Yes	Yes
Modification Attack	No	Yes	Yes	Yes	Yes
Replay attack resistance	No	Yes	Yes	Yes	Yes
Suitable for ESP8266	Yes	No	No	No	Yes

TABLE III. COMPUTATION COST COMPARISON

Registration				**Authentication**	
	User	***Server***		***User***	***Server***
Park [23]	-	$3T_H$		$3T_{PM} + 5T_H$	$3T_{PM} + 5T_H$
Our	-	$2T_H$		$2T_{PM} + 4T_H$	$4T_H$

B. Performance Evaluation

To evaluate the computational cost of our scheme in servers and clients, we define some notations as follows.

- T_{PM} : The computing time of elliptic curve scalar point multiplication;
- T_H : The computing time of secure hash function.
- - : No operations need to be performed.

In order to show the effectiveness of the proposed scheme, we compare security features and functionality with other schemes. The detail showed in Table II, the proposed model can achieve the mutual authentication between devices and trusted authority. Various kinds of issues including inside attack resistance, modification attack, and replay attack resistance and so on. Hence, considering that our scheme supports more security properties and suitable for IoT communication systems based on ESP8266. Moreover, this research aims to propose a light weight and robust security. Hence, in this section is to compare our scheme with some recently related scheme in terms of computation cost. Detailed of comparisons are show in Table III. We summarize the result of mutual authentication with key agreement phase in similar proposed protocol. The consumption is mainly determined by above unit, T_H and T_{PM}. In general, T_{PM} is the most costly execution among

these computations. The proposed model could reduce the consumption of server, and consumes less computing time on both the devices side and server side.

V. Implementation and Result

In this paper C++ and Crypto library in Arduino IDE are used for implementing the Elliptic Curve Diffie-Hellman key exchange. The environment of this experiment used Arduino Uno + ESP8266-01 shown in figure 6. ESP8266 has a 32-bit microcontroller running at 80 mhz. the sketch uses 302,700 bytes (60%) of program storage space. The maximum space is 499,696 bytes. Global variables use 31,972 bytes (39%) of dynamic memory, leaving 49,948 bytes for local variables. Maximum is 81,920 bytes.

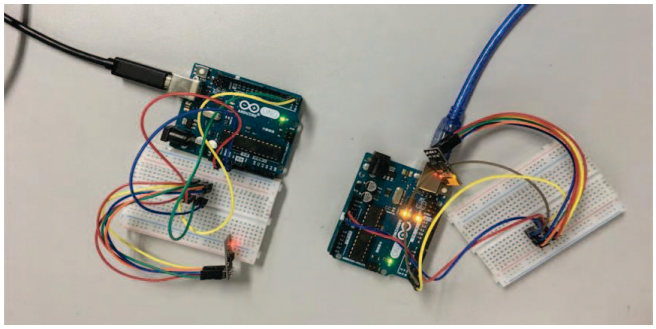

Figure 6. The hardware set-up for our experiment.

Result

Alice private key (a):
f823c0a132c69287bad33d3f7e242728b086da998f48a1cacd0c72486d8e426f
Alice public key (Y_a):
f580687eabfeda9ba2c2904b95111b54787fbc98d7eb399d9b71fb856dc5867a
Bob private key (b):
20fbb9882977442018dd1abf92bb4eb4d1ffa6174b7071bd725a1b93da2c449
Bob public key (Y_b):
bca531883a280523e69b9233f55d8ad2b8f0e8d93685fd09471ffcbfad0f874e
Alice and Bob shared secret (aY_b), (bY_a):
1b5a001e7c5a2d5f4bcc30afe34759c9f921cdf1335423dd1b577363e8704129.

After the implementation, we can see that the curve25519 key length that we implemented in the Arduino environment is 256 bits a bit longer than [7]'s scheme on the private key, public key and secret key share, but the mutual authentication with key agreement could achieve more security solution and effectively prevent exhaustive attacks. With the results, it show that our security approach is the most suitable of the ESP8266 models.

VI. Conclusion

In this paper, we proposed an efficient mutual authentication and key agreement scheme for IoT devices with low-cost module ESP8266. In order to overcome the security weaknesses of Kodali and Naikoti's scheme, we proposed Elliptic Curve Diffie-Hellman (ECDH) key exchange on curve25519 for IoT using ESP8266. Our security model can prevent the security weakness of Kodali and Naikoti's scheme which is vulnerable to various attacks such as replay attacks, modification attacks, and man-in-the-middle attacks. Our scheme not only has many security advantages but also performs efficiency in comparison to other relevant schemes. Thus, our proposed scheme is more secure and suitable for the communications between ESP8266 modules in the IoT environment.

References

[1] C. M. S. Rodrigues and B. S. L. Castro, "A vision of internet of things in industry 4.0 with ESP8266," International Journal of Electronics and Communication Engineering and Technology (IJECET), vol. 9, no. 1, pp. 1-12, 2018.

[2] J. Deogirikar and A. Vidhate, "Security Attacks inIoT: A Survey," International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 32-37, 2017.

[3] M. T. Hammi, E. Livolant, P. Bellot, A. Serhrouchni and P. Minet, "A Lightweight IoT Security Protocol," Cyber Security in Networking Conference (CSNet), pp. 1-8, 2017.

[4] J. Seo, J. Park, Y. J. Kim, D. Hwang, K. Kim, K.-H. Kim and K.-B. Lee, "An ECDH-based Light-weight Mutual Authentication Scheme on Local SIP," ICUFN, pp. 871-873, 2015.

[5] R. Goutham, G. Lee and K. Yoo, "An anonymous ID-based remote mutual authentication with key agreement protocol on ECC using smart cards," Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp. 169-174, 2015.

[6] X. Li, J. Niu, M. Z. A. Bhuiyan, FanWu, M. Karuppiah and S. Kumari, "A Robust ECC-Based Provable Secure Authentication Protocol With Privacy Preserving for Industrial Internet of Things," IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, vol. 14, no. 8, pp. 3599-3609, 2018.

[7] R. K. Kodali and A. Naikoti, "ECDH based Security Model for IoT using ESP8266," International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 629-633, 2016.

[8] E. Barker, L. Chen, A. Roginsky, A. Vassilev and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography," NIST Special Publication 800-56A, vol. 3, p. 132, April 2018.

[9] D. J. Bernstein and T. Lange, "SafeCurves: choosing safe curves for elliptic-curve cryptography," [Online]. Available: https://safecurves.cr.yp.to/. [Accessed 22 January 2017].

[10] D. J. Bernstein, "Curve25519: new Diffe-Hellman speed records," Feb 9, 2006.

[11] A. M. S. V. Darrel Hankerson, Guide to Elliptic Curve Cryptography, Springer-Verlag New York, 2004.

[12] M. Manan, "ESP8266: a breakthrough in wireless semsor networks and internet of things," International Journal of Electronics and Communication Engineering & Technology (IJECET), vol. 6, no. 8, pp. 7-11, Aug-2015.

[13] S. Arasteh, S. F. Aghili and H. Mala, "A New Lightweight Authentication and Key agreement Protocol For Internet of Things," 13th International ISC Conference on Information Security and Cryptology (ISCISC2016), pp. 52-59, September-2016.

[14] X. Fan and B. Niu, "Security of a New Lightweight Authentication and Key Agreement Protocol for Internet of Things," 9th IEEE International Conference on Communication Software and Networks, pp. 107-111, 2017.

[15] Q. Jiang, S. Zeadally, J. Ma and D. He, "Lightweight Three-Factor Authentication and Key Agreement Protocol for Internet-Integrated Wireless Sensor Networks," Special Section on Security and Privacy in Application and Services for Future Internet of Things, vol. 5, pp. 3376-3392, 2017.

[16] R. Amin, S. Islam, G. Biswas, M. Khan, L. Leng and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," Computer Network, vol. 101, pp. 42-62, 2016.

[17] T. K. Goyal and V. Sahula, "Lightweight Security Algorithm for Low Power IoT Devices," Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1725-1729, Sept-2016.

[18] "IEE Standard for identity-Based Cryptographiv Techniques using pairings," IEEE Std 1363.3, 2013.

[19] "Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Elliptic curve generation," ISO/IEC 15946-5, vol. 5, 2017.

[20] "Public Key Cryptography for the Financial Services Industry Services Industry The Elliptic Curve Digital Signature Algorithm (ECDSA)," ANS X9.62, November-2005.

[21] W. Zhang, D. Lin, H. Zhang, C. Chen and X. Zhou, "A Lightweight Anonymous Mutual Authentication with Key Agreement Protocol on ECC," 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 170-176, 2017.

[22] E. H. Teguig, Y. Touati and A. Ali-Cherif, "ECC based-Approach for Keys Authentication and Security in WSN," 9th IEEE-GCC Conference and Exhibition (GCCCE), 2017.

[23] K. Park, K. Lee and Y. Park, "Cryptanalysis and improvement of an efficient two-party authentication key exchange protocol for mobile environment," International Conference on Electronics, Information, and Communication (ICEIC), pp. 24-27, 2018.

[24] J. C. Mingping Qi, "An efficient two ‐ party authentication key exchange protocol for mobile environment," International Journal of Communication Systems, vol. 30, no. 16, pp. 1-8, 2017.