Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 9297562, 7 pages
http://dx.doi.org/10.1155/2016/9297562

*Research Article*

# Study of Wireless Authentication Center with Mixed Encryption in WSN

**Yiqin Lu, Jing Zhai, Ronghuan Zhu, and Jiancheng Qin**

*School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510000, China*

Correspondence should be addressed to Jing Zhai; 1320503196@qq.com

Received 25 March 2016; Accepted 29 May 2016

Academic Editor: Iftikhar Ahmad



WSN (wireless sensor network) has been used in a wide range of applications nowadays. Sensor networks may often relay critical data; thus, security must be a high priority. However, due to their limited computational, energy, and storage resources, sensor nodes are vulnerable to attack. So how to protect sensor nodes from attacks without raising computational capability and energy consumption is a worthwhile issue. A WAC (wireless authentication center) with mixed encryption named "MEWAC" is proposed. MEWAC is based on MCU (Microcontroller Unit) and WiFi (Wireless Fidelity) module and uses RSA, AES (Advanced Encryption Standard), and SHA-1 (Secure Hash Algorithm 1) to provide high performance authentication and data encryption services for sensor nodes. The experimental results show that MEWAC has the advantages of low cost, low power consumption, good performance, and stability; moreover, the authentication protocol improves the security of WSN and reduces the overhead in node authentication.

## 1. Introduction

The WSN has recently attracted a lot of interest due to the range of applications it enables [1]. It can be used in many applications such as battlefield, environmental surveillance, and smart home [2]. Security is the prerequisite for the application to be implemented. In order to protect the information security of WSN, it is necessary to identify nodes. But it is a challenge for the nodes to run encryption algorithms and store data due to the limited computational capability and resources.

In recent years, the research on sensor node authentication has achieved some results. User entity authentication of public key system [3] in WSN is proposed for the first time, but it brings a large amount of computation. Reference [4] proposes symmetric key encryption algorithm with low-energy consumption is required to the applicable sensor networks, but AES algorithms have difficulty managing their own keys. So the security of this protocol is yet to be further improved. Some authors propose a distributed authentication scheme [5, 6]. However, a lot of nodes are involved. The computation and communication overhead will increase with authentication requests times increasing. As a result, the energy consumption is relatively large. Reference [7] proposes authentication schemes that leverage sensor cooperation to achieve data authentication in an unattended wireless sensor network. But this scheme assumes that each node and sink node share a pair of keys, which easily cause single point failure. According to mobility of sensor node in WSN, an efficient node authentication and key exchange protocol are introduced in [8]. The protocol reduces the overhead in node reauthentication and also provides untraceability of mobile nodes.

We propose a wireless authentication center with mixed encryption named "MEWAC" according to shortcomings of the current schemes. MEWAC has the advantages of low cost, low power consumption, good performance, and stability; moreover, the authentication protocol improves the security of sensor nodes and reduces the overhead in node authentication.

The scope of this paper covers the following: Section 2 introduces the MEWAC-based WSN topology. Section 3 describes the design of MEWAC. Section 4 shows MEWAC workflow. Section 5 presents the experimental results. Conclusions and future work are given in Section 6.

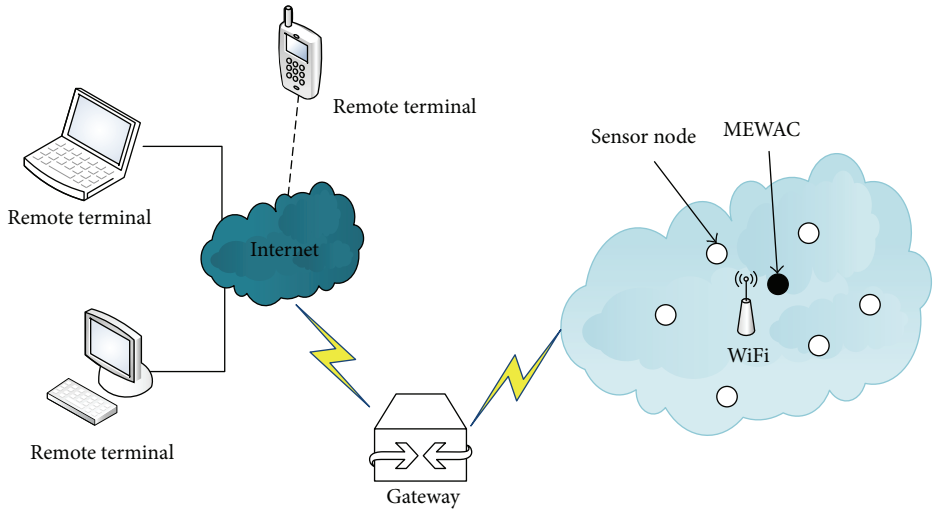


Figure 1: The MEWAC-based smart home topology.

## 2. MEWAC-Based WSN Topology

A WSN mainly consists of autonomous sensors used to collect information and to cooperatively pass their data through the network to a main location [9]. In order to simplify the application environment, we use MEWAC in smart home. Compared with the general sensor network, the number of sensor nodes in smart home is relatively small, and the smart home topology is relatively stable. So just one MEWAC can meet the actual demand. The smart home topology which is based on MEWAC is shown in Figure 1.

The topology is formed by the home gateway, sensor nodes, MEWAC, and remote terminals. The sensor nodes can send authentication requests and data requests to MEWAC. MEWAC provides authorization and data encryption storage services for sensor nodes. If the sensor node cannot be certified by MEWAC or read correct data from MEWAC, it cannot access the WSN.

## 3. Design of MEWAC

*3.1. Overall Structure.* Figure 2 shows the overall structure of MEWAC, which is the encryption product composed of hardware and software. The following sections will introduce hardware modules and software modules in detail.

*3.2. Hardware Design.* MEWAC is divided into three parts in the hardware framework: WiFi module, MCU module, and configuration module, which are shown in Figure 3. The WiFi module is used to connect to the Internet, which communicates with the MCU module via SDIO bus. The MCU module runs various algorithms and stores data needed for sensor nodes. Configuration module provides modification functions, such as ID number and keys.

In order to avoid the wireless communication link being cracked, the performance of the WiFi module must be able to meet the actual requirements. Through analysis and comparison, we select WM631-M as WiFi module. This module supports IEEE 802.11b+g standard [10]; the transmission speed can reach 54 Mbps.

In order to achieve low power consumption and low cost and maintain good function, the STM32 MCU is chosen as hardware platform. It offers a 32-bit product range that combines very high performance, real-time capabilities, digital signal processing, and low-power, low-voltage operation, while maintaining full integration and ease of development. Finally, the MCU we choose is STM32F405RGT6, which offers the full performance of the Cortex™-M4 core (with floating point unit) running at 168 MHz. This chip has not only 1 MB of FLASH and 192 KB of SRAM, but also rich peripherals.

We use UART port as configuration port, through which we can change the data stored in MCU flexibly. UART is a universal serial data bus for asynchronous communication and usually used as peripheral devices integrated in a microprocessor.

*3.3. Software Design.* It is vital to realize the function of MEWAC based on the above hardware platform. The software framework of MEWAC is shown in Figure 4, which consists of five parts: (1) MCU logic processing module; (2) WiFi module driver; (3) WiFi protocol stack; (4) security module; (5) TCP/IP protocol stack.

TCP/IP protocol stack is based on uIP 1.0 [11]. The uIP protocol stack is a very small TCP/IP protocol stack, which can be used in the embedded system built by the microprocessor. The security module is the core of MEWAC. It stores the application program and processes the authentication request and data request sent by sensor nodes. The WiFi module driver and WiFi protocol stack are based on the Linux platform source since the MCU platform is too weak to run Linux operation system.

*3.4. Extendibility.* In the hardware design of MEWAC, we set aside a large amount of storage space. Furthermore, configuration module is added, which makes it easy to change

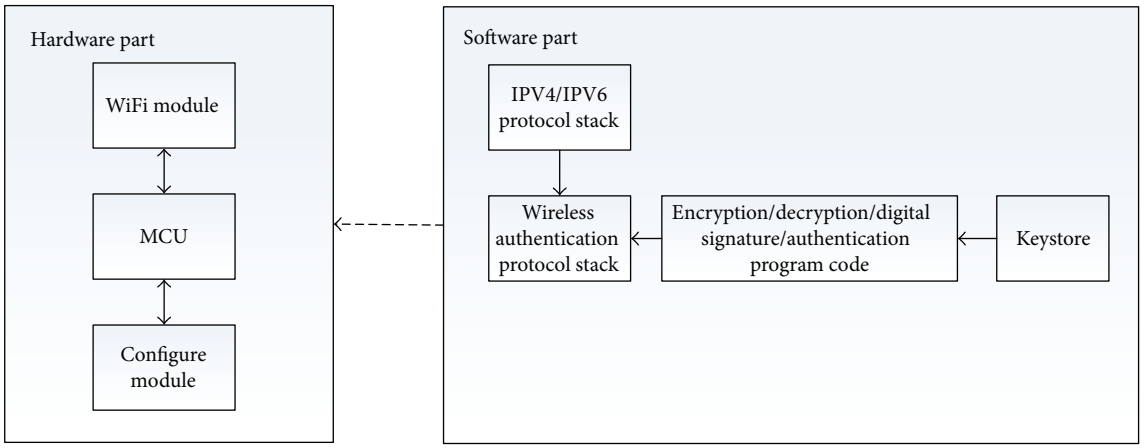


Figure 2: The overall structure of MEWAC.

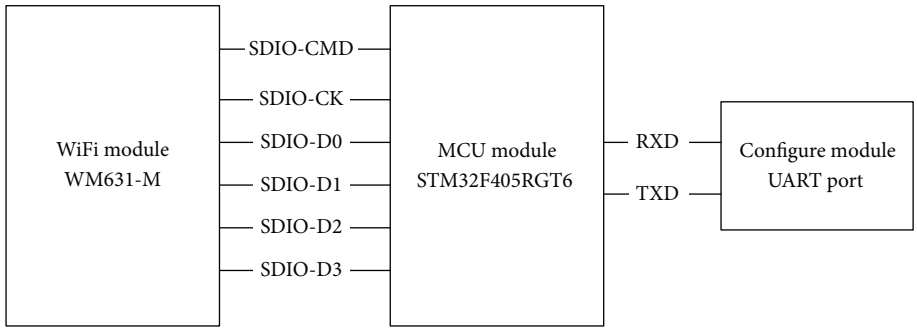


Figure 3: Hardware framework of MEWAC.

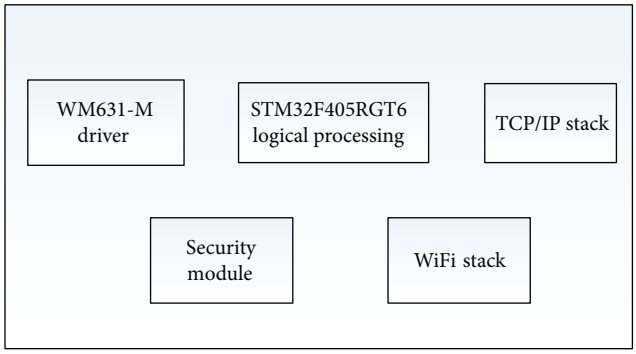


Figure 4: Software framework of MEWAC.

the data stored in MCU; in terms of program design, 8-bit space in data packet is reserved. These provide convenience for the further escalation of MEWAC.

## 4. MEWAC Workflow

*4.1. The Basic Working Process.* When users start up intelligent controller to monitor the equipment in the smart home, sensors will send authentication requests to MEWAC. After being certified by MEWAC, sensors will begin to detect the temperature, humidity, and so forth. After finishing the authentication, if the sensor nodes want to communicate with each other, sensors can send key requests to MEWAC to negotiate another AES session key, which is used to encrypt the data for communication between nodes. In addition to the key requests, the sensor can also send other data requests. It depends on the specific circumstances.

The workflow of MEWAC is shown in Figure 5. After MEWAC is powered on, it begins the initialization process immediately and reads the configuration information to access the wireless network. After that, MEWAC receives and analyzes the incoming network packets. If the incoming packet is authentication request, MEWAC will start the authentication process. If the node passes the certification successfully, it will be added to the trust list by MEWAC. MEWAC will return the AES key to the node, and the node will access the WSN. If the incoming packet is data request, MEWAC will check whether the node is in the trust list. If not, the packet will be dropped. Otherwise the packet will be decrypted, and the data required by the node will be encrypted and returned to the node.

*4.2. Wireless Authentication Protocol.* In the process of data communication, we select RAS, AES, and SHA-1 to encrypt data. RSA with asymmetric keys and AES with symmetric key are state-of-the-art cryptographic algorithms [12]. SHA-1 is used to compute a message digest for a message or data file that is provided as input [13]. The SHA-1 is computationally

Table 1: Authentication packet of sensor node.

| Product ID number | Serial number | Hash value | Reserved bits | Client ID number |
|---|---|---|---|---|
| 152 b | 192 b | 160 b | 8 b | 8 b |
| | 512 b (cipher text) | | | 8 b (plain text) |
| | | 520 b | | |

Table 2: Authentication packet of MEWAC.

| ID number | License | Random number | Hash value | Reserved bits |
|---|---|---|---|---|
| 88 b | 240 b | 16 b | 160 b | 8 b |
| | | 512 b (plain text) | | |
| | | 512 b (cipher text) | | |

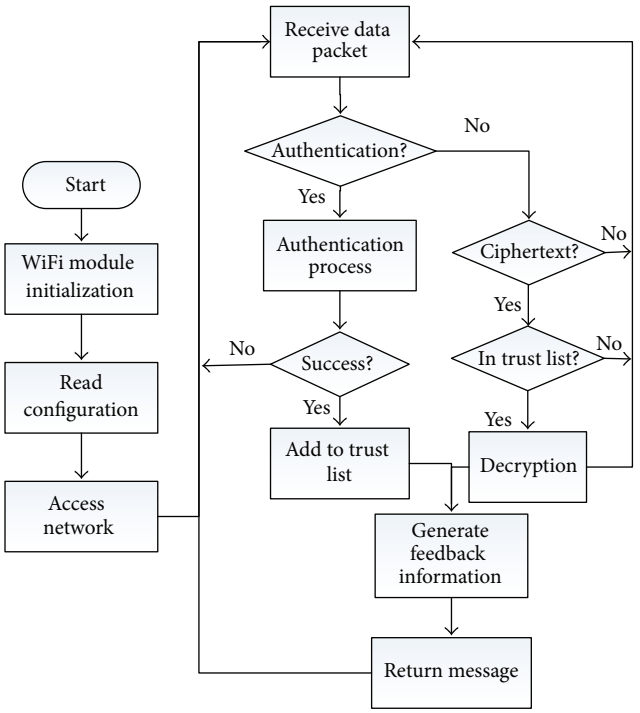


Figure 5: Workflow of MEWAC.

infeasible to find a message which corresponds to a given message digest. And any change to a message in transit will, with very high probability, result in a different message digest, so the signature will fail to be verified.

MEWAC and each node have their own information, including the ID number, public and private key. To be able to identify and communicate with each other, MEWAC stores ID number and the public key of each node which is within MEWAC's jurisdiction, and each node stores ID number and the public key of MEWAC. All of the information stored in memory is read-only. The public key of each device is unique. If it is fake, which device it is can be known from the public key, so as to realize the purpose of tracking the system.

As shown in Table 1, the authentication packet sent by sensor node includes its own product ID number, client ID number, serial number, and hash value. The sensor nodes within the jurisdiction of MEWAC have the same product ID number. To distinguish between various sensor nodes, each node is assigned to a client ID number. The serial numbers mainly composed of random numbers can prevent the message from becoming intercepted and retransmitted. The hash value is calculated by the product ID number and the serial numbers, which can effectively prevent the message from being tampered with during the process of transmission. We use the public key of MEWAC to encrypt the 512-bit plain text. The data encrypted by public key can only be decrypted by the corresponding private key, which is saved locally to ensure that the message cannot be decrypted by others.

The structure of the authentication packet sent by MEWAC is shown in Table 2. The license is calculated according to the serial number of node. We use the public key which corresponds to the sensor node's ID number to encrypt the packet. Just like the authentication packet of sensor node, the authentication packet of MEWAC also has the effect of antiretransmission, antitampering, and anticrack.

Authentication process is shown in Figure 6. The authentication between sensor node and MEWAC is bidirectional. In order to prevent DoS (Denial of Service) attacks, we limit how many times the sensor node can visit MEWAC in a unit of time.

*4.3. Encrypted Communication.* RSA algorithms easily manage their own keys. However, their computing quantity is large and computing time is relatively long. AES algorithms are just the opposite. The combination of both has the salient features of Symmetric Cryptography, having fast speed and being easy to process, and features of Asymmetric Cryptography such as being secured, avoiding key transportation, and providing the power to the users to generate their own keys of variable length [13]. As a result, during the authentication process, MEWAC and the node make agreement to use part of the returned message as the first communication AES key. The license and random number are used as the AES-256-bit key. After finishing the authentication process, sensor nodes can send data requests encrypted by the AES-256-bit key to WDongle.

*4.4. Performance Analysis of Mixed Encryption.* Compared to the traditional method of using AES encryption for sensor

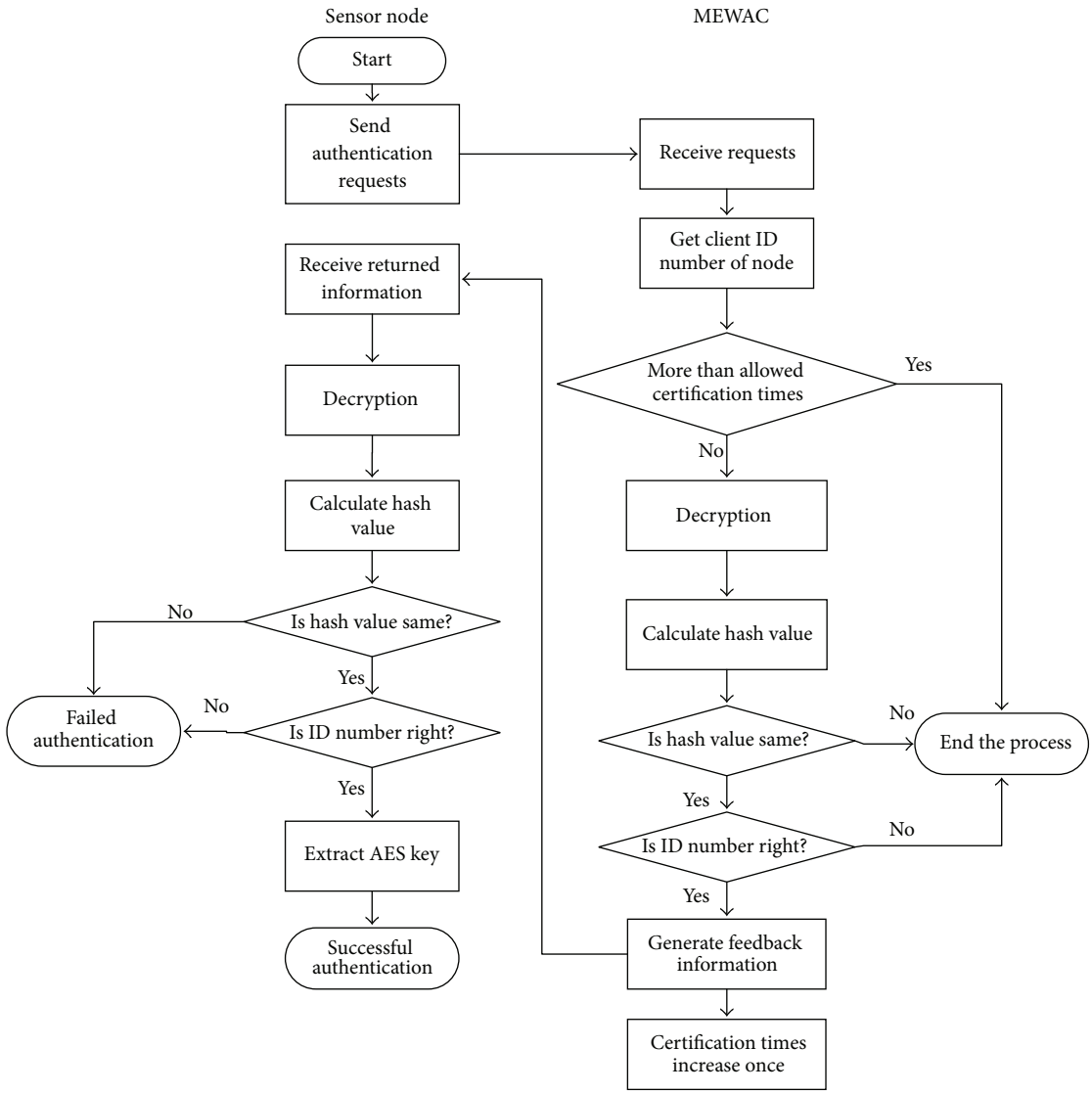


Figure 6: Authentication process of MEWAC.

nodes authentication [4], the mixed encryption method has a higher security. Firstly, SHA-1 encryption algorithm can ensure the integrity of the data; then RSA encryption packet is used to pass a AES-256 b session key, which avoids the insecurity of key transportation. Although the energy consumption of mixed encryption is relatively larger than that of AES encryption, the increase in energy consumption is not large, which is in acceptable range of the nodes and MEWAC. On the one hand, the shorter the key length is, the less the energy consumption is [14]. So we choose RSA-512 b which is relatively shorter; on the other, RSA encryption packet which also is the authentication package is not only used to transfer the AES key, but also used for identity authentication. So there is no extra information for the transmission of the AES key.

Calculation and energy consumption of mixed encryption are smaller compared to the RSA encryption for nodes authentication, and the node security can also be guaranteed.

Table 3: Hardware parameters and costs.

| | |
|---|---|
| MCU | STM32F405RGT6 |
| Storage | 1 MB |
| Voltage | 5 V |
| Frequency | 168 MHz |
| NIC | WM631-M |
| Memory | 192 KB |
| Power | 1.4 W |
| Price | 6.4$ |

## 5. Experiments

According to MEWAC's design principle, we implement this WAC based on MCU and WiFi module, which is used as our experimental platform. To test and approach the pure

Table 4: Algorithms running time.

| Encryption model | | RSA | | | AES | | SHA-1 |
|---|---|---|---|---|---|---|---|
| Key length | 512 b | 768 b | 1024 b | 128 b | 256 b | | |
| Time | 602 ms | 2.01 s | 4.89 s | 185 $\mu$s | 610 $\mu$s | | 227.9 $\mu$s |

Table 5: Average response time.

| | Authentication process | Data request process |
|---|---|---|
| Time | 1.544 s | 13.1 ms |

performance of MEWAC, we use 9 clients on the personal computer instead of 9 sensor nodes. The hardware parameters and costs are shown in Table 3. We can see it complies with the requirements of low power and low cost. According to the results, the entire design occupies about 49 KB memory including 14 KB used by memory management and occupies about 211 KB storage volume. Therefore, the MEWAC also has the possibility of further escalation.

Different encryption methods directly affect the performance of WEWAC and nodes, such as security level and power consumption. On the one hand, a longer key consumes more power and results in more heat dissipation [14]; on the other, the security of any type of cryptography depends on number of bits used in a key and the amount of computation required to break the cipher [15]. According to the test result and the comparison of encryption algorithms, we chose RSA-512 b, AES-256 b, and SHA-1. The running time which is taken by these algorithms to encrypt the same 512-bit packets in our platform is shown in Table 4.

We stipulate the whole times a sensor node can send authentication request to MEWAC in 15 minutes are not more than 12. In fact, times the sensor node can visit MEWAC in a unit of time can be artificially set according to the actual situation. During the experiment, the sensor can only operate in a limited range. If out of WiFi coverage, the sensor cannot access wireless network to communicate with MEWAC.

In order to know the average response time of authentication and data requests process in our platform, we do some tests and the experimental results are shown in Table 5. Authentication process is the whole process shown in Figure 6. Data requests process includes mainly data request transmission, reception of MEWAC's feedback packet, and correct reading of packet.

In order to known the stability of MEWAC, a stress test is carried out. The sensor node sends 500 times of data read request to the WAC with AES session key. The success rate of this test is 100%. Figure 7 shows the response time for each data request processing, which indicates that the WAC has sufficient performance and stability.

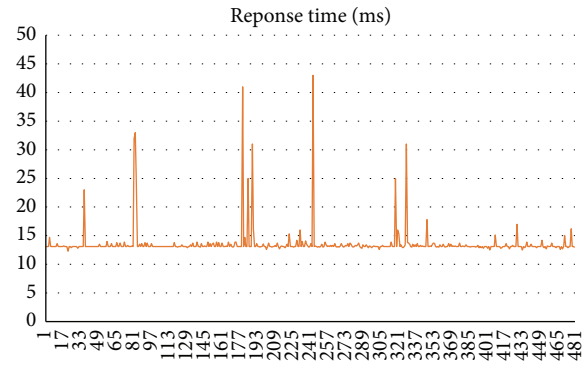


Figure 7: Stress test.

## 6. Conclusions and Future Work

The authentication for resource-constrained sensor node identity can effectively protect sensor node from attacks. This paper proposes a mixed encryption wireless authentication center named MEWAC according to shortcomings of the existing schemes. MEWAC can reduce the overhead and prevent retransmission, tampering, and DoS attacks. Furthermore, we can use the unique public key of each device to track the system, and the sensor nodes can only work in a limited range which the WiFi of MEWAC can cover. The experimental results show that MEWAC not only has the advantages of low power consumption, low cost, and good performance, but also provides advanced protection for sensor nodes.

MEWAC is extensible, which provides convenience for the further upgrading. In future work, we can consider using Bluetooth instead of WiFi, which can reduce the cost and power consumption of MEWAC ulteriorly [16]. We also can use steganography [12, 17, 18] to provide more advanced security protection for sensor nodes.

MEWAC can be generalized to common WSN. Multiple MEWACs collaborate to complete the node certification, which forms a distributed authentication scheme. Except for being used in WSN, the MEWAC can protect the copyright of software.

## Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

## References

[1] E. Sabbah, A. Majeed, K.-D. Kang, K. Liu, and N. Abu-Ghazaleh, "An application-driven perspective on wireless sensor network security," in *Proceedings of the 2nd ACM International Workshop on Quality of Service & Security for Wireless and Mobile Networks (Q2SWinet '06)*, pp. 1–8, ACM, October 2006.

[2] X.-Y. Yang, Q. Zhang, and L.-X. Wei, "A robust entity authentication in wireless sensor networks," in *Proceedings of the International Conference on Information Engineering and Computer Science*, pp. 1–4, IEEE, Wuhan, China, December 2009.

[3] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, ACM Press, Washington, DC, USA, October 2004.

[4] H. Lee, K. Lee, and Y. Shin, "Implementation and performance analysis of AES-128 CBC algorithm in WSNs," in *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10)*, pp. 243–248, IEEE Press, 2010.

[5] K. Bauer and H. Lee, "A distributed authentication scheme for a wireless sensing system," *ACM Transactions on Information and System Security*, vol. 11, no. 3, pp. 1–35, 2008.

[6] R. Bellazreg, N. Boudriga, and M. Hamdi, "A dynamic distributed key tunneling protocol for heterogeneous wireless sensor networks," in *Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1077–1082, IEEE, Liverpool, UK, June 2012.

[7] R. D. Pietro, C. Soriente, A. Spognardi, and G. Tsudik, "Collaborative authentication in unattended WSNs," in *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec '09)*, pp. 237–244, Zurich, Switzerland, March 2009.

[8] K. Han, K. Kim, and T. Shon, "Untraceable mobile node authentication in WSN," *Sensors*, vol. 10, no. 5, pp. 4410–4429, 2010.

[9] A. Q. Zhao, Y. N. Weng, Y. Lu, and C. Y. Liu, "Research on dynamic routing mechanisms in wireless sensor networks," *The Scientific World Journal*, vol. 2014, Article ID 165694, 7 pages, 2014.

[10] Y.-Q. Lu, D.-W. Wu, and J.-C. Qin, "Wireless authentication center based on embedded Wi-Fi technology," *WIT Transactions on Information and Communication Technologies*, vol. 59, pp. 387–394, 2014.

[11] Y.-Q. Lu, D.-W. Wu, and J.-C. Qin, "Design of embedded web server based on MCU and IPv6," *International Journal of Advancements in Computing Technology*, vol. 5, no. 5, pp. 1232–1240, 2013.

[12] S. F. Mare, M. Vladutiu, and L. Prodan, "Secret data communication system using steganography, AES and RSA," in *Proceedings of the IEEE 17th International Symposium for Design and Technology of Electronics Packages (SIITME '11)*, pp. 339–344, October 2011.

[13] D. Eastlake III and P. Jones, *US Secure Hash Algorithm 1 (SHA1)*, RFC Editor, 2001.

[14] A. Kakkar, M. L. Singh, and P. K. Bansal, "Comparison of various encryption algorithms and techniques for secured data communication in multinode network," *International Journal of Engineering and Technology*, vol. 2, no. 1, pp. 87–92, 2012.

[15] W. Heng-Qing and S. Ru-Min, "The security of public key cryptosystem depends on the length of the key," *Scientific and Technological Information: Academic Research*, vol. 34, 2008.

[16] C. You-ping, "Study on the coexistence of bluetooth and WiFi," *Science and Technology Wind*, no. 8, p. 99, 2009.

[17] N. Hopper, L. von Ahn, and J. Langford, "Provably secure steganography," *IEEE Transactions on Computers*, vol. 58, no. 5, pp. 662–676, 2009.

[18] J. Daemen and V. Rijmen, "The first 10 years of advanced encryption," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 72–74, 2010.