Sensors and Materials, Vol. 34, No. # (2022) 1–##
MYU Tokyo

S & M ####

Fast Elevator Vibration Signal Cloud Collection System Using Data Compression and Encryption Algorithms

Hsiung-Cheng Lin,* Fu-Yu Chou, Yu-Xiang Hong, and Yi-Wei Wang

Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

(Received October 3, 2021; accepted January 31, 2022; online published April 4, 2022)

Keywords: Wi-Fi module, database, elevator, compression, encryption

Currently, commercial elevators still lack the capability of collecting dynamic vibration data. This makes it difficult to generate an early warning sign for preventive maintenance or predict possible elevator malfunction in advance. For this reason, we aim to develop a wireless elevator vibration data collection system through data compression and encryption. First, the vibration signal produced from an accelerometer in the elevator is received and compressed into hexadecimal data by a microprocessor. Second, the received data is encrypted on the basis of Advanced Encryption Standard (AES), and it is then sent to a server via a Wi-Fi module. Third, through decryption and decompression processes, the original vibration data is restored in a MySQL database that can be tracked from an online web human–machine interface (HMI). Experimental results demonstrated that the restored data is consistent with the original data. Additionally, the total time for data transmission with compression and encryption is reduced fourfold compared with the traditional method. This verifies the superiority of the proposed system in terms of speed, robustness, and accurate data transmission.

1. Introduction

An elevator is a machine that can vertically transport people or freight between floors of a building, a vessel, or other structures. It is a major transportation tool to access different locations of buildings in the modern world.(1) Vibration signals generated from the elevator operation are the most influential factor in determining the stability and safety of elevators.(2) With the increasing application of Internet of Things (IoT), an elevator integrated with IoT may provide an effective way to solve the current elevator safety problems. For instance, data collection, data transmission, central processing, and a software program together can constitute a complete elevator IoT monitoring system. Therefore, the real-time effective supervision and maintenance of elevators can be realized by elevator operation data analysis using a human–machine interface (HMI) platform.(3–6)

Elevator fault detection using profile extraction and deep feature extraction was previously proposed.(7) Time-series data was extracted from start and stop events, and informative deep features were used in fault detection from acceleration and magnetic signals. However, this approach may not be applicable in real applications owing to the lack of an IoT operation for

*Corresponding author: e-mail: hclin@ncut.edu.tw

ISSN 0914-4935

easily tracking collected data. Another study introduced the application of elevator IoT technology in real-time monitoring, fault diagnosis, alarm, and maintenance.[8] Unfortunately, only a schematic system design was given with a lack of more detailed techniques. Moreover, an elevator ride comfort monitoring method using smart phones was proposed. Although the experimental results indicated that it was stable, economic, and convenient, the data transmission rate of smart phones is limited,[9] meaning that such a system may be unsuitable for the collection and transmission of large data sets.

Data compression can effectively reduce the space consumed on a disk and thus increase the data transmission speed. Data compression has many applications such as in energy efficiency, electromyography, medical images, and hyperspectral imaging.[10–19] To ensure data security, encryption can prevent databreaches, regardless of whether or not the data is in transit or at rest.[20–23] After transforming data into ciphertext, the data can be turned back into plaintext by using the decryption key. As above, the integration of data compression and encryption has become an important issue, especially when data is transmitted to the cloud via the internet.[24] For this reason, in this paper, we combine data compression and encryption algorithms to achieve a fast elevator vibration signal transmission to the cloud.[25]

2. System Structure

As shown in Fig. 1, the proposed elevator vibration signal collection system mainly comprises 1. an elevator side, consisting of (a) an accelerometer, (b) a microcontroller, and (c) a Wi-Fi module; 2. a cloud side, consisting of (a) a server and (b) an HMI. These parts are described in further detail as follows.

Elevator side:

a. Accelerometer: An LSM6DSR accelerometer is used for measuring the elevator vibration level in the *X*-, *Y*-, and *Z*-axes. It is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope with an extended full-scale range for the gyroscope of up to 4000 dps.
b. Microcontroller: A PIC24FJ256GB106 microprocessor is used. It has an 8 MHz internal oscillator; 10-bit, up to 16-channel analog-to-digital (A/D); and three analog comparators with a programmable input/output configuration. It works as a core controller, which can be used for data compression and encryption processes. All processed data can be sent to the Wi-Fi module via a universal asynchronous receiver/transmitter (UART) for transmission.

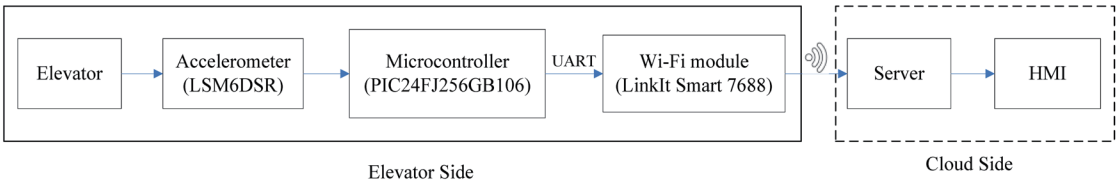


Fig. 1. (Color online) System structure.

c. Wi-Fi module: A LinkIt Smart 7688 Wi-Fi module is used to transmit the vibration signal after compression and encryption. Its key features are as follows: it (1) supports the Wi-Fi and USB host and SD cards, and (2) pins out for PWM, I2C, SPI, UART, Ethernet, and I2S.

Cloud side:

d. Server: The server decrypts and decompresses data, and then sends it to a MySQL database.
e. HMI: The HMI is designed to view the vibration data from the elevator.

2.1 Process of elevator side

A flowchart of the elevator-side process is shown in Fig. 2. The process is described in more detail as follows.

a. Check if Wi-Fi is connected. If yes, go to the next step. Otherwise, continue the same procedure.
b. Open UART.

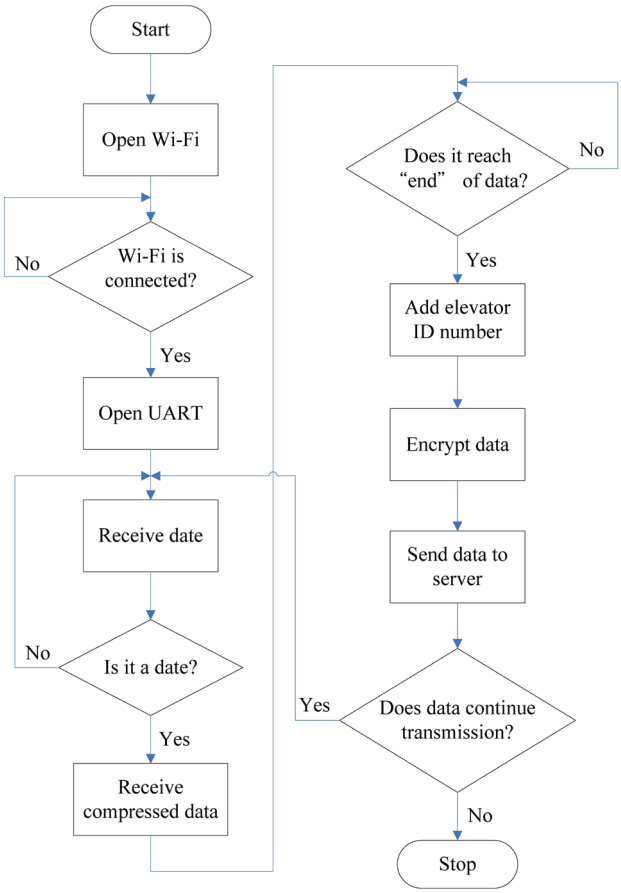


Fig. 2. (Color online) Flowchart of elevator-side process.

c. Receive date and check if it is a date. If yes, go to the next step. Otherwise, continue the same procedure.
d. Receive compressed data.
e. Check if the data reaches “end”. If yes, go to the next step. Otherwise, continue the same procedure.
f. Add elevator ID to the data.
g. Encrypt data using AES.
h. Send data to the server.
i. Check if the data transmission is continuing. If yes, go to Step c. Otherwise, the system stops.

The data before encryption is shown in Fig. 3. The elevator ID number is placed in the front, followed by the collection time and the data. The ID number consists of two letters plus nine decimal numbers. The collection time has 14 decimal numbers including year (four numbers), date (four numbers), hour (two numbers), min (two numbers), and s (two numbers). The transmitted data comprises two hexadecimal numbers. In the example below, we have ID number et001123456; collection time 20210623142514 (year, date, hour, min, s); data \x32\xC9\x92…; end: transmission done.

2.1.1 Data compression and decompression

(1) Data compression

The data compression process shown in Fig. 4 includes two stages: a decimal process and a binary process. In the first stage, first input decimal data such as *X*-, *Y*-, and *Z*-axis elevator vibration (mg) and cushion pressing distance (mm). Second, operate the mathematical computation flowchart shown in Fig. 4(a). Finally, output the data with the decimal code. Note that the cushion pressing distance is used to measure the elevator load weight due to their linear relationship. In the second stage, first input the output decimal code from the first stage, which is converted to binary code as the input data. Second, carry out the binary logic process in Fig. 4(b). Finally, the result is obtained as an array type, i.e., arr[0]–arr[5], in hexadecimal code. Using the proposed data compression algorithm, the original input data can be assigned to 27–36 arrays, where each array has a capacity of one byte. After compression, the output data generates only six arrays, i.e., arr[0]–arr[5].

(2) Data decompression

The data decompression process is shown in Fig. 5. Firstly, the hexadecimal code, i.e., arr[0]–arr[5], that is obtained from the compression process is input to the data decompression

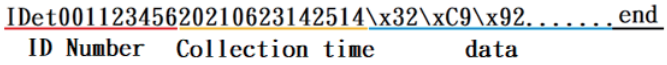


Fig. 3. (Color online) System structure.

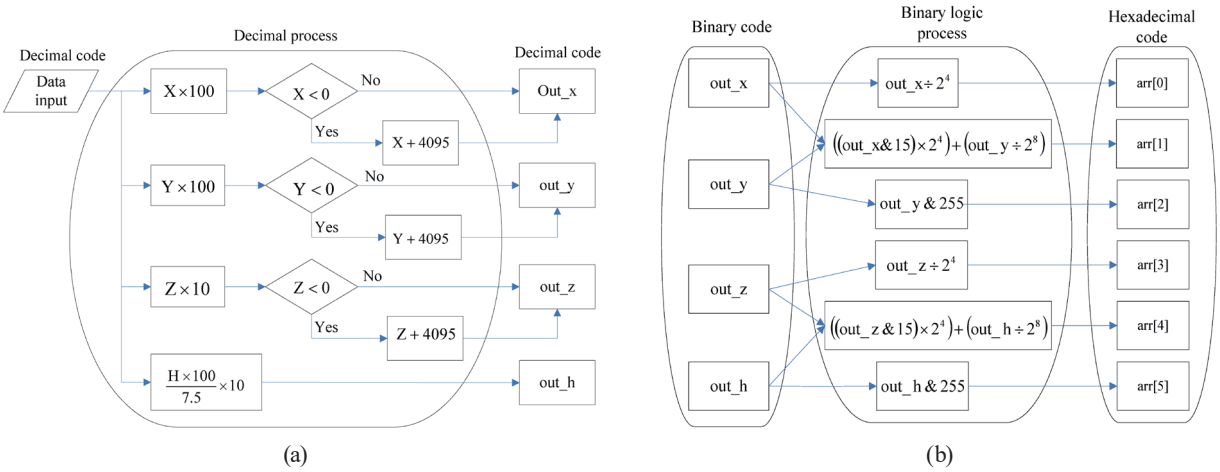


Fig. 4. (Color online) Compression processes in the (a) first and (b) second stages.

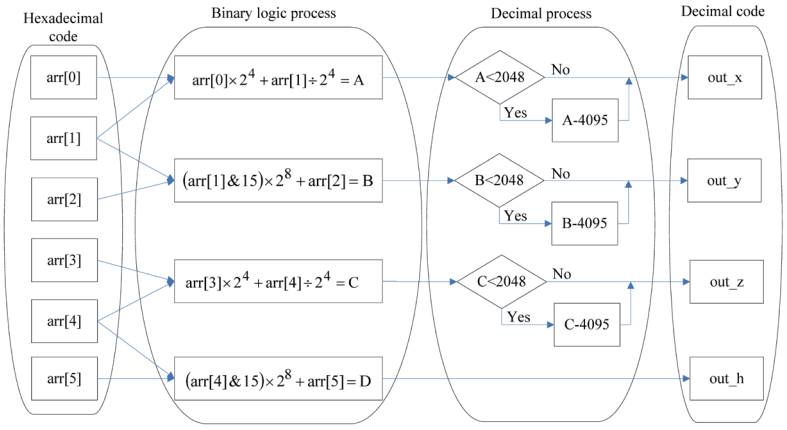


Fig. 5. (Color online) Decompression process.

model. Secondly, the binary logic process is performed according to the mathematical equations used to calculate A, B, C, and D. Thirdly, the decimal process is carried out for A, B, C, and D based on the judgment equation. Finally, the output, i.e., out_*x*, out_*y*, out_*z*, and out_*h*, expressed in decimal code is obtained. From this process, the original data such as *X*-, *Y*-, and *Z*- axis elevator vibration (mg) and cushion pressing distance (mm) can be recovered.

2.1.2 Data encryption and decryption

The data encryption process shown in Fig. 6 is based on Advanced Encryption Standard (AES) using a 128 bit symmetric-key algorithm. Each round comprises four subprocesses with a 4 × 4 byte array: 1. Substitute bytes (SubBytes). 2. Shift rows (ShiftRows). 3. Mix columns (MixColumns). 4. Add round key (AddRoundKey).

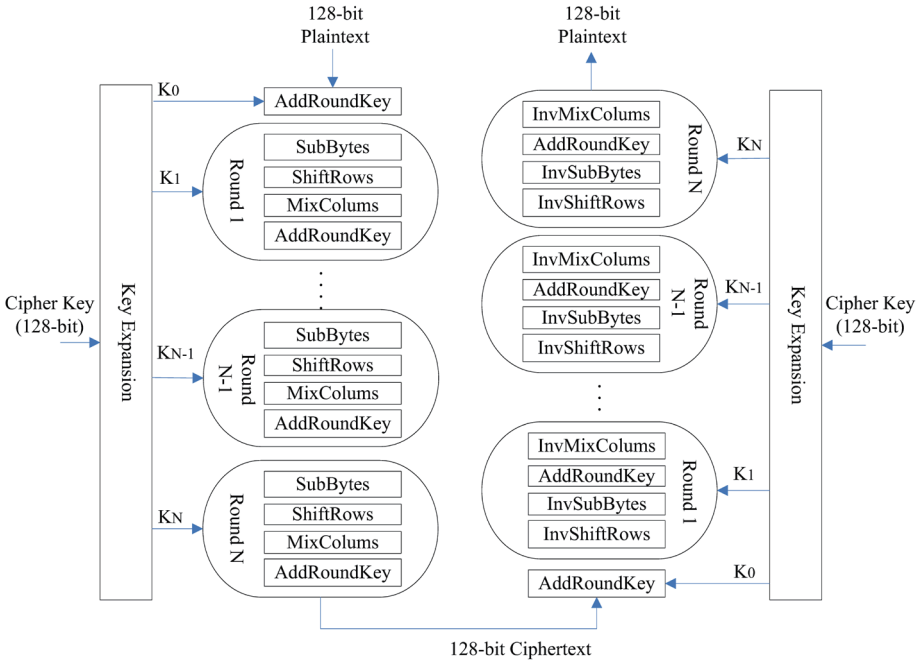


Fig. 6. (Color online) AES algorithm.

(1) Byte substitution (SubBytes)

There are 16 input bytes to be substituted with a lookup table. The result is a matrix of four rows and four columns. The SubBytes phase involves splitting the input into bytes and passing each byte through a substitution box.

(2) ShiftRows

Each of the four rows of the matrix is shifted to the left. A transposition step shifts the last three rows of the state cyclically with a certain number of steps. Each row of the 128-bit internal state of the cipher is shifted.

(3) MixColumns

By applying a linear mixing operation to the columns of the state, the MixColumns function provides diffusion by mixing the input. Each column of four bytes is transformed by taking as the input the four bytes of one column and outputting four completely new bytes, which replace the original column.

(4) AddRoundKey

The 16 bytes of the matrix are considered as 128 bits, and each byte of the state is combined with a byte of the round key using bitwise XOR. If it is the last round, the output is the ciphertext.

Otherwise, the resulting 128 bits are interpreted as 16 bytes, and another similar round is performed.

The process of decryption in AES ciphertext is similar to the encryption process but in the reverse order. Each round consists of the four processes conducted in the reverse order: 1. Add round key. 2. Mix columns. 3. Shift rows. 4. Substitute bytes.

2.2 Process of cloud side

The structure on the cloud side is shown in Fig. 7. In the receiving end, three actions are performed: 1. Receive encrypted and compressed data. 2. Decrypt and decompress data. 3. Send data to a MySQL database. The web HMI allows the user to select the desired data file based on the date and time. The data is then displayed in a sheet or graph according to the user's preferences.

2.2.1 Receiving end

The flowchart of the receiving end is shown in Fig. 8. The process is described in more detail as follows.

a. Open server for operation.
b. Receive data from the elevator side.
c. Check if data is received. If yes, go to the next step. Otherwise, return to Step b.
d. Decrypt data via the decryption key.
e. Read elevator number and check if it is correct. If yes, go to the next step. Otherwise, go to Step g.
f. Decompress data and then send data to the MySQL database.
g. Check if continuing to receive data. If yes, go to Step b. Otherwise, go to the next step.
h. End system operation.

2.2.2 Web HMI

The front page of the web HMI is shown in Fig. 9. It asks the user to select the desired data with the specific date and time. The data can be displayed in a sheet or graph. The flowchart of the web HMI is shown in Fig. 10. More details are illustrated as follows.

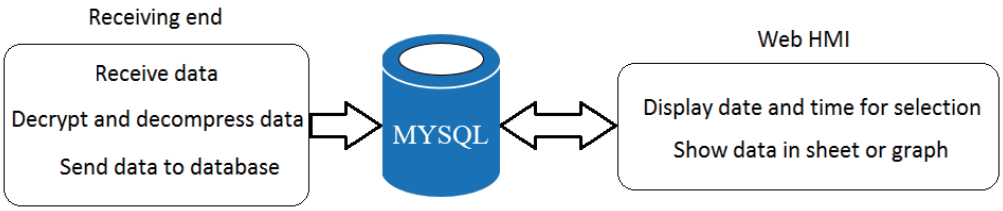


Fig. 7. (Color online) Structure on cloud side.

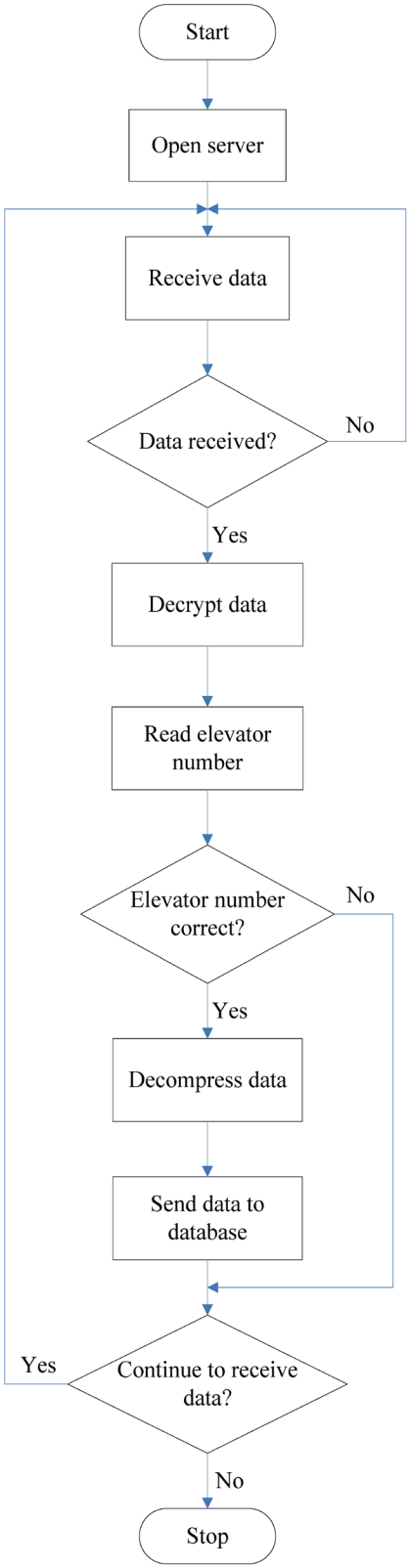


Fig. 8. (Color online) Flowchart of receiving end.

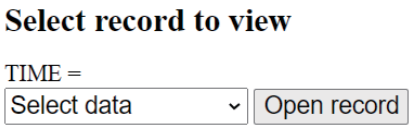


Fig. 9. Front page of web HMI.

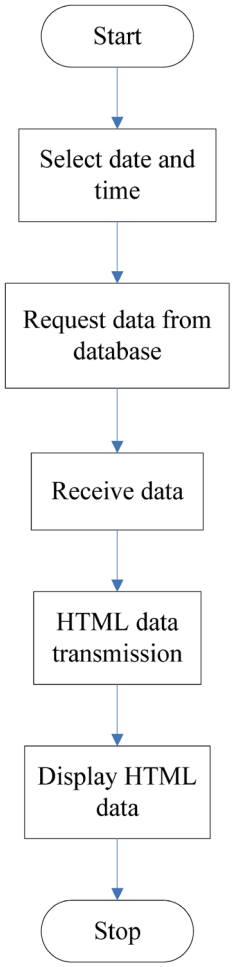


Fig. 10. (Color online) Flowchart of web HMI.

a. Select date and time for data display.
b. Request data from the MySQL database.
c. Receive data.
d. Transmit HTML data to web.
e. Display HTML data.
f. End system operation.

3. Experimental Results

On the elevator side, the compression and encryption data is exhibited from the model test. The crucial process involves the following steps: First, the vibration signal is compressed with the date and time. Second, the elevator number is added to the data. Third, the data is encrypted using AES and transmitted to the cloud side. On the cloud side, the encrypted data is decrypted and then decompressed to restore its original value. Afterward, the restored data is stored in a MySQL database.

3.1 Data compression and encryption on elevator side

The vibration signals on the elevator side before compression are shown in Table 1, which contains vibration values in three axes, i.e., the *X*-, *Y*-, and *Z*-axes. Note that the sampling period for every signal acquisition is 6 ms. For reasons of space, only 15 samples are shown in Table 1.

On the basis of the vibration signals in Table 1, the compressed data with the acquisition date and time is compressed into a hexadecimal format, as shown in Fig. 11(a). The elevator number is then added in front of the compressed data, as shown in Fig. 11(b). The encrypted data is displayed in binary code, as shown in Fig. 11(c). The data transmission time is shown in Table 2. The UART was used for the data transmission and the baud rate was set as 115200.

Table 1
Vibration values in three axes (add weight ratio).

Item no.	*X*-axis (mg)	*Y*-axis (mg)	*Z*-axis (mg)
1	0.43	−1.2	0.7
2	−2.79	4.22	−3.2
3	0.56	3.24	0.9
4	−3.22	0.92	0.6
5	0.01	1.66	−2.9
6	1.17	1.23	−1.3
7	−1.45	0.56	−0.7
8	−0.53	0.01	0.0
9	−0.53	0.56	0.8
10	0.19	0.19	0.0
11	−0.65	2.08	0.2
12	1.17	0.98	0.8
13	−0.78	1.41	0.6
14	−0.65	2.14	−0.3
15	0.07	2.14	−0.2

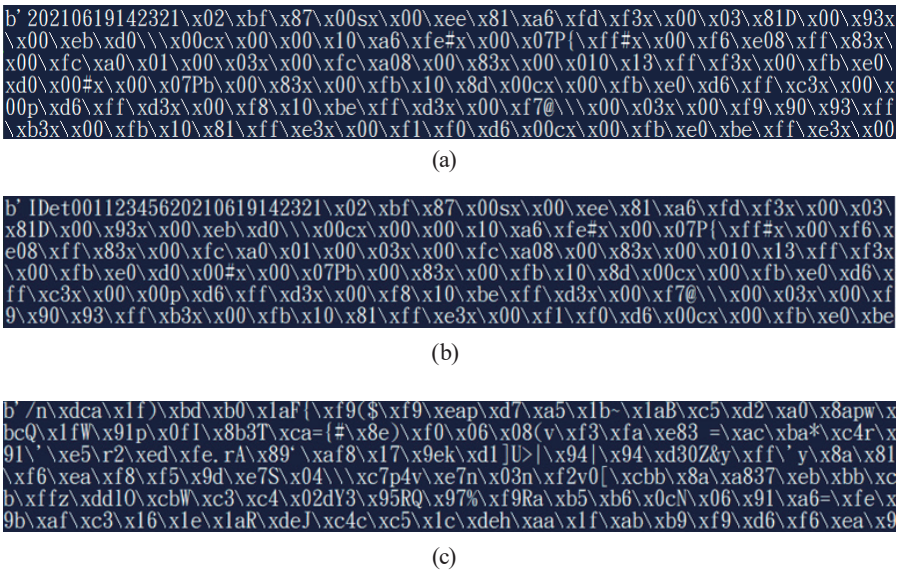


Fig. 11. (a) Compressed data. (b) Compressed data plus elevator number. (c) Encrypted data.

Table 2
Data transmission time.

Data type	Data size (byte)	Transmission time (ms)
Original data	36	3.21
Original data	27	2.34
Decompressed data	6	0.52

3.2 Data decryption and decompression on cloud side

The encrypted data received on the cloud side is shown in Fig. 12, which is confirmed to be the same data as in Fig. 11(c), indicating the success of data transmission and receipt. The data after decryption is shown in Fig. 13, which contains the elevator number, date, and time as compressed data with hexadecimal values. The data is confirmed to be the same as in Fig. 11(b), indicating the success of decryption.

The decompressed data stored on the cloud side is converted into decimal data to be stored in the MySQL database. Note that only the first 20 data are shown in Fig. 14, where (1), (2), and (3) denote *X*-, *Y*-, and *Z*-axis data, respectively. From Table 3, it can be seen that the decompressed data is consistent with the original data, indicating the success of data restoration.

3.3 MySQL database and web HMI

3.3.1 MySQL database

The data is stored in a MySQL database, where (1)–(4) denote (1) the date and time, (2) the total number of data, (3) the form name, and (4) the *X*-, *Y*-, and *Z*-axis vibration data in units of mg.

Fig. 12. Received data.

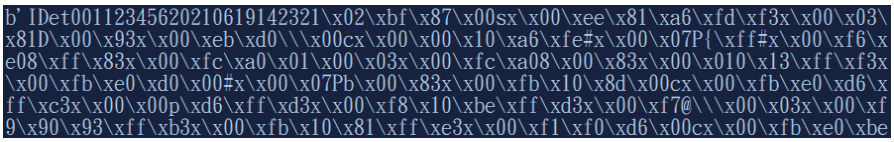


Fig. 13. Decrypted data.

(1) (2) (3)

```
0.43 -1.2 0.7
-2.79 4.22 -3.2
0.56 3.24 0.9
-3.22 0.92 0.6
0.01 1.66 -2.9
1.17 1.23 -1.3
-1.45 0.56 -0.7
-0.53 0.01 0.0
-0.53 0.56 0.8
0.19 0.19 0.0
-0.65 2.08 0.2
1.17 0.98 0.8
-0.78 1.41 0.6
-0.65 2.14 -0.3
0.07 2.14 -0.2
-1.26 1.9 -0.2
-1.39 0.92 0.0
-1.02 1.47 -0.4
-0.78 1.29 -0.1
-2.24 2.14 0.6
```

Fig. 14. Decompressed data.

Table 3
Vibration valuesin three axes.

Item number	*X*-axis (mg)		*Y*-axis (mg)		*Z*-axis (mg)	
	Original	Decompressed	Original	Decompressed	Original	Decompressed
1	0.43	0.43	−1.2	−1.2	0.7	0.7
2	−2.79	−2.79	4.22	4.22	−3.2	−3.2
3	0.56	0.56	3.24	3.24	0.9	0.9
4	−3.22	−3.22	0.92	0.92	0.6	0.6
5	0.01	0.01	1.66	1.66	−2.9	−2.9
6	1.17	1.17	1.23	1.23	−1.3	−1.3
7	−1.45	−1.45	0.56	0.56	−0.7	−0.7
8	−0.53	−0.53	0.01	0.01	0.0	0.0
9	−0.53	−0.53	0.56	0.56	0.8	0.8
10	0.19	0.19	0.19	0.19	0.0	0.0
11	−0.65	−0.65	2.08	2.08	0.2	0.2
12	1.17	1.17	0.98	0.98	0.8	0.8
13	−0.78	−0.78	1.41	1.41	0.6	0.6
14	−0.65	−0.65	2.14	2.14	−0.3	−0.3
15	0.07	0.07	2.14	2.14	−0.2	−0.2

3.3.2 Web HMI

The web HMI was designed for users to track the data records in the database. The inquiry interface is shown in Fig. 15. The major instructions include (1): Input web address, e.g., 120.108.10.131/search. The time menu "TIME" can then be opened. (2): Click "Select data" to choose data for the desired date and time. (3): Click "Open record" to view the data in a sheet or graph.

The selected data displayed in a sheet is shown in Fig. 16, which includes *X*-, *Y*-, and *Z*-axis vibration data. Note that only seven data are displayed in the figure for reasons of space.

Alternatively, the vibration data can be displayed in a graph. The *X*-, *Y*-, and *Z*-axis vibration data for the descending elevator are shown in Figs. 17(a)–17(c), respectively. There is no obvious variation of the vibration amplitude in the *X*- and *Y*-axes during the elevator operation. On the other hand, in the *Z*-axis, the data indicates that a significant vibration occurred as the elevator was descending. From 5 to 8 s, the vibration abruptly exhibited a negative value of up to −50 mg when the elevator accelerated. It then returned to zero when the elevator kept a constant speed without accelerating between 8 and 10 s. From 10 to 12.5 s, the vibration abruptly changed to a positive value of up to 70 mg when the elevator decelerated until it fully stopped.

The *X*-, *Y*-, and *Z*-axis vibration data of the ascending elevator are shown in Figs. 18(a)–18(c), respectively. Similarly to above, the vibration amplitude in the *X*- and *Y*-axes showed no distinct change during elevator operation. In contrast, in the *Z*-axis, an apparent vibration appeared when the elevator was rising. From 6 to 8.5 s, the vibration abruptly exhibited a positive value of up to 50 mg when it accelerated. It then returned to zero as the elevator remained at a constant speed without accelerating between 8.5 and 11 s. From 11 to 13 s, the vibration abruptly took a negative value of up to −71 mg while the elevator decelerated until it fully stopped.

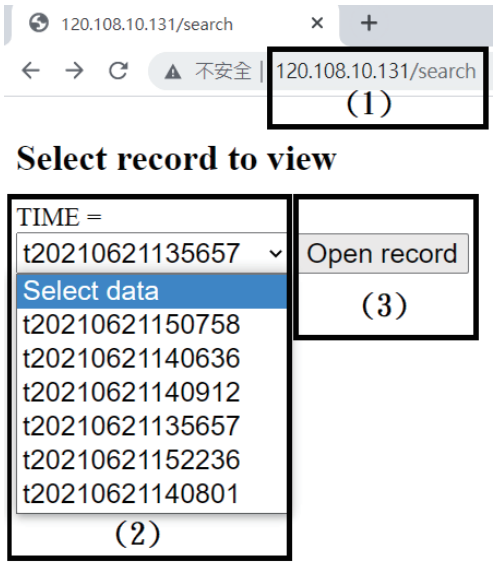
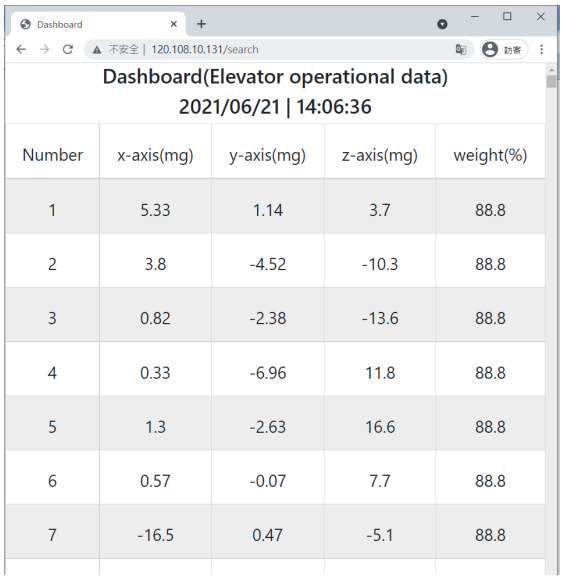


Fig. 15. (Color online) Inquiry interface.



Number	x-axis(mg)	y-axis(mg)	z-axis(mg)	weight(%)
1	5.33	1.14	3.7	88.8
2	3.8	-4.52	-10.3	88.8
3	0.82	-2.38	-13.6	88.8
4	0.33	-6.96	11.8	88.8
5	1.3	-2.63	16.6	88.8
6	0.57	-0.07	7.7	88.8
7	-16.5	0.47	-5.1	88.8

Fig. 16. (Color online) Elevator vibration data.

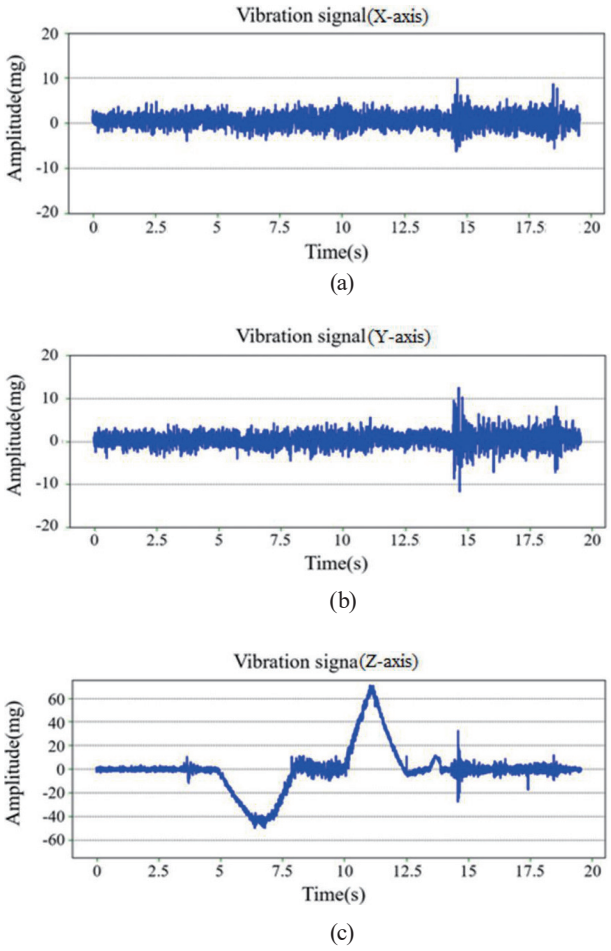


Fig. 17. (Color online) (a) *X*-axis vibration signal, (b) *Y*-axis vibration signal, and (c) *Z*-axis vibration signal for elevator descending.

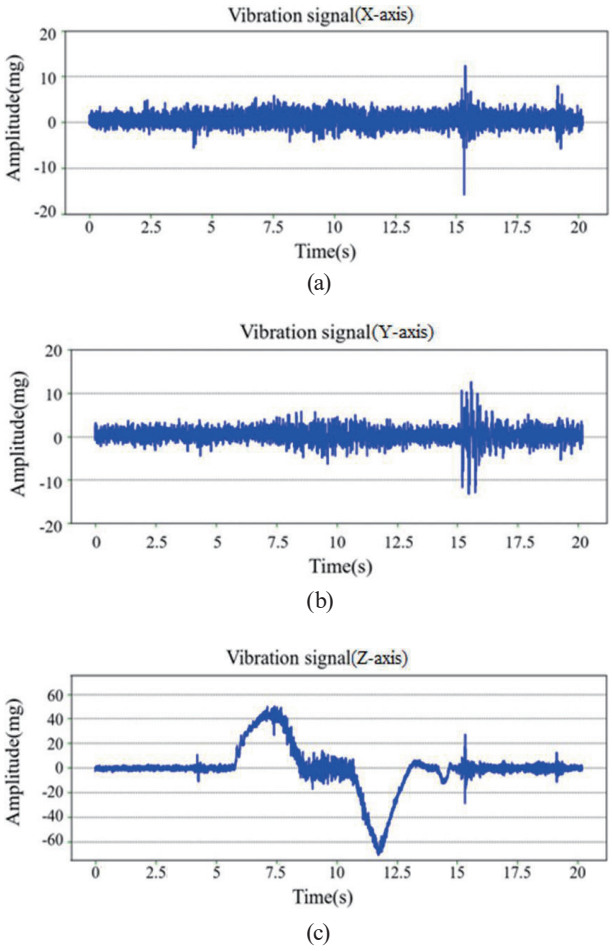


Fig. 18. (Color online) (a) *X*-axis vibration signal, (b) *Y*-axis vibration signal, and (c) *Z*-axis vibration signal for elevator rising.

4 Conclusions

In this paper, a fast wireless elevator vibration signal collection system based on data compression and encryption processes has been proposed. The experimental results confirm that original data from the elevator can be recovered with a shorter time than the data obtained by the traditional method after decompression and decryption processes. The major features of the system are as follows.

a. All collected data produced by the accelerometer on the elevator side can be sent to the cloud using a Wi-Fi module via UART.
b. The cloud side can receive real-time vibration signals in the *X*-, *Y*-, and *Z*-axes.
c. The hexadecimal compression and encryption processes can make data transmission faster and securer than the traditional method. The amount of transmission data can be reduced fourfold by compression.

d. A HMI can provide users with on-line vibration signals as either a data table or graphics.
e. The collected vibration data can be used for early warning or checking the elevator operation status as part of maintenance.

References

1 C. S. Hsu and D. J. Huang: Environ. Moni. Asse. **186** (2014) 2941. https://doi.org/10.1007/s10661-013-3591-7
2 S. H. Choi, J. S. Kim, and T. S. Kim: J. Kor. Mul. Soc. **16** (2013) 226. https://doi.org/10.9717/kmms.2013.16.2.226
3 Y. Zhou, K. Wang, and H. Liu: Proc. Com. Sci. **131** (2018) 541. https://doi.org/10.1016/j.procs.2018.04.262
4 X. Pan: Inter. J. Sma. Hom. **10** (2016) 183. https://doi.org/10.14257/ijsh.2016.10.12.17
5 Z. Ming, S. Han, Z. Zhang, and S. Xia: Inter. J. On. Bio. **14** (2018) 121. https://doi.org/10.3991/ijoe.v14i08.9179
6 S. Li, P. Hui, and J. Yan: Adv. Mat. Res. **926** (2014) 1253. https://doi.org/10.4028/www.scientific.net/amr.926-930.1253
7 K. Mishra and K. Huhtala: Appl. Sci. **9** (2019) 2990. https://doi.org/10.3390/app9152990
8 Y. Zhou, K. Wang, and H. Liu: Pro. Com. Sci. **131** (2018) 541. https://doi.org/10.1016/j.procs.2018.04.262
9 Y. Zhang, X. Sun, X. Zhao, and W. Su: Mec. Syst. Sig. Pro. **105** (2018) 377. https://doi.org/10.1016/j.ymssp.2017.12.005
10 J. Spiegel, P. Wira, and G. Hermann, 2018 IEEE INDIN (2018). 447. https://doi.org/10.1109/INDIN.2018.8471921
11 G. Biagetti, P. Crippa, L. Falaschetti, A. Mansour, and C. Turchetti: Sensors. **21** (2021) 5160. https://doi.org/10.3390/s21155160
12 C. Itiki, S. Furuie, and R. Merletti: BioM. Eng. On. **13** (2014). https://doi.org/10.1186/1475-925X-13-25
13 I. Suarjaya: Inte. J. Adv. Com. Sci. App. **3** (2012) 14.
14 G. Xin and P. Fan: Sci. Rep. **11** (2021) 12372. https://doi.org/10.1038/s41598-021-91920-x
15 C. Kwan and J. Larkin: J. Sig. Inf. Pro. **10** (2019) 96. https://doi.org/10.4236/jsip.2019.103007
16 C. Kwan, J. Larkin, B. Budavari, B. Chou, E. Shang, and T. D. Tran: Computers **8** (2019) 32. https://doi.org/10.3390/computers8020032
17 S. Parikh, D. Ruiz, H. Kalva, G. Fernández-Escribano, and V. Adzic: IEEE J. Bio. Hea. Inf. **22** (2018) 552. https://doi.org/10.1109/JBHI.2017.2660482
18 G. Xin, Z. Li, Z. Zhu, S. Wan, P. Fan, and K. Letaief: IEEE Com. Let. **25** (2021) 798. https://doi.org/10.1109/LCOMM.2020.3035595
19 H. Pan, Y. Lei, and C. Jian: EURASIP J. Ima. Vid. Pro. **142** (2018). https://doi.org/10.1186/s13640-018-0386-3
20 Z. Han, S. Huang, H. Li, and N. Ren: The Electron. Lib. **34** (2016) 471. https://doi.org/10.1108/EL-09-2014-0158
21 Z. Cai and D. Huang: Com. Mea. Con. **25** (2017) 241.
22 Y. Sun and X. Wang: Pac. Eng. **38** (2017) 194.
23 L. Liu, S. Xiao, L. Zhang, M. Bi, Y. Zhang, J. Fang, and W. Hu: Opt. Com. **398** (2017) 62. https://doi.org/10.1016/j.optcom.2017.04.015
24 S. Marcelo, A. Jorge, F. Andres, G. Jose, O. Sergio, and G. Alberto: Neural Comput. Appl. **28** (2017) 953. https://doi.org/10.1007/s00521-016-2258-zM.
25 Jia, X. Gao, H. Li, and H. Pang: Shoc. Vibra. **2021** (2021) 1. https://doi.org/10.1155/2021/4547030