Artificial Intelligence Model Based Security Protection Method for IoT Applications

Xiaolong Luo[1], Xiaoli Chen[2(✉)], Jie Wei[1], Liang Zhang[2], Luping Xu[2], and Bijun Zhao[2]

[1] Zhejiang Water Conservancy Information Publicity Center, Hangzhou 310009, China
[2] Zhejiang Ponshine Information Technology Co., Ltd., Hangzhou 311100, China
chenxiaoli@ponshine.com

Abstract. The issue of model privacy security is increasingly affecting the application systems of Artificial Intelligence Internet of Things (AI-IOT) terminals, where it is challenging to protect the privacy of the underlying AI models. In this paper, we propose a security protection RC6-plus algorithm based on cryptography and access control for AI model security in IoT applications. Specifically, the proposed method effectively protects the privacy of crucial algorithms in the program by encrypted storing the model parameters, as well as storing and code obfuscating the neural network structure and parameters of the AI model independently while adding the isolation treatment of the JNI communication layer. The results of the experiments verify the effectiveness of the proposed method.

Keywords: Model privacy security · internet of things · artificial intelligence model · encryption technology · access control

1 Introduction

With the continuous development of society and technology, Internet of Things (IoT) technology has been widely used in various fields. The IoT is gradually becoming an indispensable part of people's life and work because of its intelligence and connectivity. In recent years, IoT and Artificial Intelligence (AI) technologies are becoming more and more closely integrated, and more and more AI technologies are being applied to IOT end devices. Such devices also have the ability of offline recognition, which can realize various applications in highland, deep sea, remote areas, archaeology, exploration, and geological examination [4,5,10].

In the plateau, deep sea, and other particular environments, conventional networking equipment may have the problem of unstable network transmission, leading to interruption in the data transmission process. The Artificial Intelligence

X. Chen—This work was funded by the Zhejiang Provincial Department of Water Resources Science and Technology Plan Project, Zhejiang, China (Project no. RC2238).

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 573, pp. 143–157, 2024.
https://doi.org/10.1007/978-3-031-65126-7_15

Internet of Things (AI-IOT) offline identification device can realize real-time processing and identification of data by deploying the model on the body of the device, and the data processing method that does not depend on the transmission environment reduces the risk of data transmission interruption and solves the problem of its online real-time identification instability, which has high practical value. In archaeology, exploration, and geological examination, AI-IOT can process and comprehensively analyze a large amount of data. Based on the characteristics of offline processing technology, this AI-IOT can also work autonomously through unitized design and intelligent scheduling technologies to further improve work efficiency. Conventional monitoring systems may have signal interruptions and analysis errors in bad weather, such as rain, wind, and lightning. At the same time, AI-IOT can deploy AI models on the equipment to realize real-time monitoring and grasp the comprehensive situation of the machine, environment, and other parameters, and conduct intelligent analysis and processing of this, increasing the application areas of artificial intelligence [8,21]. At the same time, the application of AI-IOT is becoming increasingly widespread and will face many challenges, including physical security, identity identification issues, external attacks, vulnerability issues, data validation, model security, program update mechanism, and communication security. In particular, AI models involve a large amount of data and privacy; once attacked and maliciously changed, it is easy to affect the stability and reliability of the whole system, and people start to pay attention to the impact of read and write operations of smart IoT terminal device data on the privacy and security of AI models. Therefore, how to protect the AI models of IoT terminals has become a critical research direction [1,20,21,23].

This paper is organized as follows: Sect. 2 reviews the current research status of AI-IOT; Sect. 3 reviews an efficient AI model application system architecture for IoT terminals and crucial algorithm research (RC6, RC6-plus); Sect. 4 focuses on the research content experimental results and analysis process. Finally, conclusions are drawn in Sect. 5.

2 Related Work

At present, AI model security protection for IoT terminals in academia and industry is actively carrying out relevant research, mainly around the following aspects:

1. Data security protection
   For the security threats and risks faced by the data security of IoT terminals, researchers have proposed a series of data security protection techniques. For example, encryption technology is used for secure data storage, integrity protection, and access control mechanisms [6,8,10,21,23].
2. AI model protection
   In response to the security risks faced by AI models, researchers have developed a series of protection techniques for AI models. For example, AI models are encrypted, compressed, and cut to increase the security of the models; a reliable AI algorithm framework is used to ensure the reliability of model training and inference [1,13,20,23,26].

3. Encryption algorithm research
   Encryption algorithms protect secure communication between IoT endpoints and AI models. Researchers have recently researched encryption algorithms to ensure data security during the communication process. For example, researchers have proposed trusted authentication techniques, secure transmission, and data encryption algorithms [2,3,7,15,18,19,24,25].
4. Multi-level security mechanism design
   In order to enhance the security between IoT terminals and AI models, the researchers proposed a multi-level security mechanism design. This approach will strengthen security in several aspects, such as model management, model usage, and model storage, to maximize the security of the IoT terminal system [2,7,9,11,12,14,16–18].

The above are only some of the research contents related to the research of AI model security protection methods for IoT terminals. However, the research in this area is still in its initial stage, and further in-depth research and exploration are needed in many aspects. Therefore, it is of significant theoretical and application value to research the AI model security protection method for IoT terminals [7,11–13,19,24].

Based on this, this research will draw on domestic and international research results and methods to explore a more comprehensive, detailed, and feasible AI model security protection method from various aspects, such as data security, model protection, and encryption algorithms. The research aims to solve the challenges faced by AI model security on IoT endpoints, protect the security and privacy of devices and data, and provide useful academic and practical references for the innovative development of related fields [2,3,7,18,25].

This paper first introduces the basic concepts of IoT and AI models and analyzes the existing AI model security threats and the limitations of existing solutions. Secondly, the paper also proposes a security protection method based on encryption and access control and details the implementation process and related technical details of the method.

Finally, the paper verifies the effectiveness and feasibility of the proposed method through experiments. The experimental results show that the method has high security and scalability and can provide more comprehensive protection for AI models in IoT applications.

Overall, the application of AI-IoT offline identification devices can effectively improve the efficiency of data sampling, reduce the cost of manual work, enhance accuracy, and is suitable for data collection and processing tasks in various complex environments. The development of this technology plays an essential role in developing the modern manufacturing industry, promoting industrial upgrading, and enhancing national strength. The research results of this paper are of great significance to the development and popularization of IoT applications and provide a helpful reference for AI safety.

The main contributions of this work can be summarized as:

1. An efficient system architecture of AI model is proposed for IoT terminals;

2. A new improved RC6 (Rivest cipher 6) algorithm is proposed for enhanced model encryption, denoted as RC6-plus;
3. An improved design scheme between AI-IOT software layers is proposed to further enhance security.

3 Proposed Method

This section focuses on two parts; the first part is an explanation of the model training steps and the architecture diagram of the AI-IOT model inference system; the second part introduces the traditional RC6 while proposing the RC6-plus algorithm and its improvement process; the reader will learn about the method of AI model security protection research for IoT terminals described in this paper.

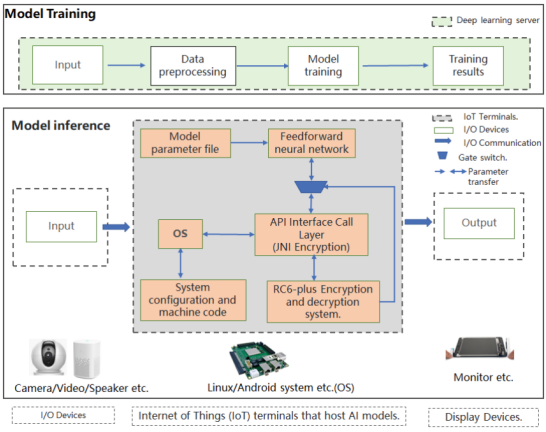

Fig. 1. Server model training and AI-IOT model inference system architecture

3.1 System Architecture Design

As the architecture is shown in Fig. 1, the model training task is done on the deep learning GPU server, the model inference is made on the smart IoT terminal, and the model can run on Linux or Android operating systems. The basic steps of model training (feedback neural network) include data input, data preprocessing, feedback neural network calculation, and parameter storage. Among them, model inference (feedforward neural network) mainly includes input, model parameter loading, and model computation inference. The data on the input side can come from the camera, microphone, or locally stored data, and the output receiver can be the display or stored in the database. This paper focuses on the software system architecture approach to the smart IoT terminal. The operating system (OS, Operate System) extract is coded by the connected terminal machine and

transmitted to RC6-plus for encryption and decryption via JNI. If the secret key is decrypted successfully, the gate control valve (GS, Gate Switch) is opened, allowing the data received by the I/O to reach the feedforward neural network layer through the JNI layer to complete the inference, which is often accompanied by the process of model loading. The final result is again transmitted to the terminal display device through the I/O interface.

3.2 Key Algorithm Study

In this section, we analyze the traditional RC6 algorithm in detail and find that RC6 has certain defects in the application of this project [7,9,22]. Based on this, we propose the RC6-plus encryption algorithm with flexible control of the number of round bits and successfully apply it to our intelligent IoT terminal AI project, achieving good results.

The detailed process of the RC6 algorithm has the following steps:

1. RC6 is one of the AES candidate algorithms. It is an improved version of the RC5 algorithm. (w, r, b) in RC6-$w/r/b$ denotes the operation word length, the number of iteration rounds, and the user master key length, respectively. Usually, we choose the arithmetic word length w = 32 bits (bit). The plaintext packet length is 4 characters (128 bits). RC6 consists of input encryption, r rounds of iteration, and output transformation.
   Input encryption:

$$(A, B, C, D) = (A, B + Str(0), C, D + Str(1)) \tag{1}$$

   Round r iteration:

$$t = [B * (2B + 1)] <<< lg(w); \tag{2}$$

$$u = [D * (2D + 1)] <<< lg(w); \tag{3}$$

$$A = [A \oplus t <<< u] + Str(2i); \tag{4}$$

$$C = [C \oplus u <<< t] + Str(2i + 1); \tag{5}$$

$$(A, B, C, D) = (B, C, D, A); \tag{6}$$

   Output transformation:

$$(A, B, C, D) = (A + Str(2r + 2), B, C + Str(2r + 3), D); \tag{7}$$

2. In the algorithm of RC6, Str(i) represents the subkey word, while "<<<" and ">>>" represent the controlled left rotation and right rotation, respectively. The symbol controls the amount of rotation, followed by the number of rotations is controlled by the lowest 5 bits of the number following the symbol. In addition, to meet the fixed grouping bit length requirement, RC6 also uses a quadratic function B*(2B+1) to strengthen the diffusion property, which is very different from most other encryption and decryption algorithms. The graphical representation of the wheel function is shown below (in Fig. 2), where $f(x)$ represents the following nonlinear invertible function:

$$f(x) = (x * (2x + 1)) <<< lg(w) \tag{8}$$

3. RC6 is a high-performance, highly flexible group iterative cipher whose compact and transparent architecture makes it widely used in monolithic microcontrollers. In addition, RC6 performs even better in application scenarios such as fingerprint recognition and POS machines. The data-dependent cyclic nature of RC6 can significantly improve encryption efficiency while its memory requirements are relatively low, and the highly integrated internal cache technology can significantly reduce production costs.
Although the RC6 algorithm is designed for simplicity and efficiency, it still has some shortcomings, such as the lack of performance of the nonlinear function f because the bit diffusion of f is unidirectional. The diffusion speed is slow, and the average computation is w/2 = 16 additions due to the use of multiplication in f, so the nonlinear function becomes the bottleneck of the operation speed. In addition, RC6 has significant differences in the encryption and decryption algorithms, which is also a drawback. To address these issues, some improvements were made to RC6 as follows.

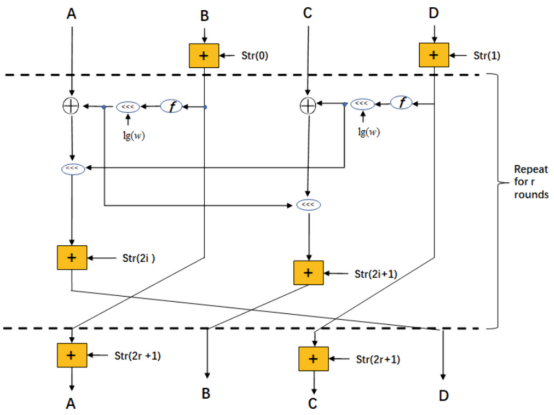

Fig. 2. Principle of RC6 algorithm wheel function

The detailed process of RC6-plus algorithm improvement has the following steps:

1. Adjust the number of rounds r and w-bit word length to balance security and algorithm performance
   First, we assume the original RC6 algorithm uses r-rounds and w-bit word lengths. For each word A, B, C, D, we can calculate its output using the following equation:

$$A' = ((A \oplus B) <<< s) \oplus K_0 \tag{9}$$

$$B' = ((B \oplus C) <<< t) \oplus K_1 \tag{10}$$

$$C' = ((C \oplus D) <<< u) \oplus K_2 \tag{11}$$

$$D' = ((D \oplus A) <<< v) \oplus K_3 \tag{12}$$

   where $\oplus$ denotes the XOR operation, $<<<$ denotes a circular shift, K_i is a round constant, and s, t, u, and v are parameters that need to be calculated based on the w-bit word length. The formulae for these parameters are as follows:

$$w = 32 \rightarrow r = 20, s = 7, t = 2, u = 13, v = 8 \tag{13}$$

$$w = 64 \rightarrow r = 20, s = 35, t = 5, u = 31, v = 16 \tag{14}$$

   Suppose we want to improve the algorithm to increase its performance. In that case, we can reduce the number of rounds r or decrease the w-bit word length to reduce the amount of computation for encryption. However, this will also reduce the security of the algorithm.
2. Modify the generation method of the RC6-plus algorithm wheel constant K_i
   Increase the randomness and complexity of wheel constant generation to enhance the strength of the encryption algorithm. The wheel constant Ki of the RC6 algorithm is derived from a specific key. If this key can be guessed or leaked, then the security of the encryption is threatened. Therefore, we need to enhance the randomness and complexity of the wheel constant generation to improve the strength of the algorithm. We use more complex key derivation algorithms or introduce more wheel constants to increase the randomness and complexity of encryption.
3. Optimization of operations in the encryption wheel
   The original RC6 algorithm uses relatively simple arithmetic, so we must introduce more complex nonlinear functions and improve the algorithm using iso-or and circular shifts. Simple attack methods can break this simple arithmetic, so we can optimize the arithmetic in the encryption wheel by introducing more complex nonlinear functions to increase the strength of the algorithm.
   With the above three improvements, we can improve the security and performance of the RC6 algorithm to make it more suitable for practical applications, and the improved RC6 algorithm is noted as RC6-plus. Suppose the RC6-plus algorithm uses r rounds of encryption, with 4 inputs A, B, C, D,

and 4 round constants K_i in each round, and 4 outputs A', B', C', D', then the computation process of the round function can be expressed as

$$B \leftarrow B + K_i \tag{15}$$

Pass the new value B calculated in Eq. 1 into the f function:

$$D \leftarrow D + f(B, C) \tag{16}$$

$$D \leftarrow D <<< s \tag{17}$$

$$D \leftarrow D \oplus B \tag{18}$$

The new value D is calculated and rounded with C:

$$C \leftarrow C + K_i + 1 \tag{19}$$

$$C \leftarrow C <<< t \tag{20}$$

$$C \leftarrow C \oplus D \tag{21}$$

Immediately afterward, the new values C and D are rounded with A:

$$A \leftarrow A + f(C, D) \tag{22}$$

$$A \leftarrow A <<< u \tag{23}$$

$$A \leftarrow A \oplus C \tag{24}$$

After following the above cryptographic round, a new set of output values is obtained:

$$A' \leftarrow A, B' \leftarrow B, C' \leftarrow C, D' \leftarrow D \tag{25}$$

where $i = 1, 2, ..., r, s, t, u$ are the computed parameters, and f(x, y) denotes the improved RC6-plus algorithm. A nonlinear function is introduced in the program for converting inputs to outputs. Specifically, the function $f(x, y)$ can be defined as

$$f(x, y) = (x \oplus y) <<< ((x \cdot y) \mod w) \tag{26}$$

This function is a nonlinear function that converts the inputs B and C into an output word D. The class Similarly, another nonlinear function $f'(x, y)$ can be defined as

$$f'(x, y) = (x \oplus y) <<< (w - ((x \cdot y) \mod w) \tag{27}$$

This function converts inputs C and D into an output word A to further disrupt data flow during encryption. The above is the basic structure schematic of the wheel function in the RC6-plus algorithm, in which more complex nonlinear functions are introduced to enhance the security of the encryption algorithm. In practical applications, the wheel function can be adjusted and optimized according to specific needs to improve the strength and performance of the encryption algorithm. RC6-plus is similar to encryption and decryption, which is not repeated here in this paper, and the wheel function of the RC6-plus algorithm can be viewed as shown in Fig. 3.

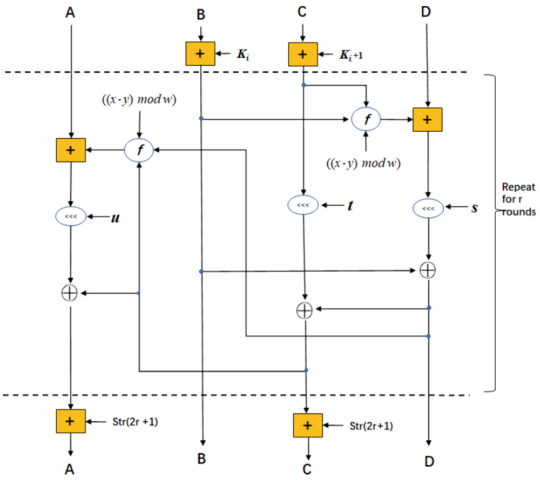

Fig. 3. Principle of wheel function of RC6-plus algorithm

Table 1. Raspberry Pi 4 Model B configuration parameters

Configuration items	Specification
CPU	Broadcom BCM2711, quad-core Cortex-A72(ARM v8), 64-bit SoC at 1.5 GHz
Memory	LPDDR4 SDRAM, 4 GB, MicroSD card slot
Network	Gigabit Ethernet, dual-band 802.11ac wireless card (with optional Bluetooth 5.0)
USB	2 * USB 3.0 and 2 * USB 2.0 ports
Video/Audio Output	2 * micro-HDMI ports supporting up to 4K resolution
Audio	Stereo output, stereo MIC, support Bluetooth audio output
GPIO	40 * GPIO pins
Operating System	Raspberry Pi OS (Debian-based operating system)

4 Experimental Results and Analysis

In this work, we tested the performance of the proposed scheme on a real Raspberry Pi-based IoT device. To evaluate the performance of the proposed scheme, we exclude the time consumed on the communication channel as it heavily depends on the network traffic. The experimental setup focuses only on the performance tests for the time used for encryption, RC6-plus operation, and decryption. The RC6-plus instances are all running on the latest Raspberry Pi (Raspberry Pi 4 Model B) with the system parameters configured, as shown in Table 1.

The following experiments are based on improved RC6 cryptographic algorithms with different bit cell lengths (64bit 208bit), using six datasets provided, each testing ten RC6 algorithms with different key lengths. The datasets are derived from the homebrew program Automatic Random Sequence, COCO dataset, ImageNet, CIFAR, MNIST, PASCAL VOC, SQuAD, Labeled Faces in the Wild, UCI Machine Learning Library, and with the addition of some data

Table 2. Performance comparison of encryption algorithms for text and digital datasets

Key Length	Average encryption time (ms) of Dataset - Algorithm			
	Text - RC6	Text - RC6-plus	Digital - RC6	Digital - RC6-plus
64bit	305	208	127	89
80bit	354	247	145	107
96bit	405	283	167	128
112bit	453	319	191	148
128bit	510	368	220	173
144bit	575	408	250	198
160bit	635	453	283	227
176bit	701	499	322	257
192bit	778	543	362	289
208bit	864	602	407	323

from web crawlers. In this experiment, 1000 copies of each data type were sampled randomly from these original data sets as the original data for the experiments in this paper. The average encryption time of each data type is taken after the experiment. The test data will be listed in a table, where each row represents a test key length, and each column represents the encryption time and encryption strength of each test item in the dataset.

In this paper, we experimentally compare the encryption time and strength of the traditional RC6 algorithm and RC6-plus algorithm on text, digital, image, and audio datasets, as shown in Table 2, Fig. 4, Table 3, and Fig. 5.

Table 3. Performance comparison of encryption algorithms for image and audio datasets

Key Length	Average encryption time (ms) of Dataset - Algorithm			
	Image - RC6	Image - RC6-plus	Audio - RC6	Audio - RC6-plus
64bit	1896	1186	273	191
80bit	2162	1389	317	224
96bit	2447	1610	361	256
112bit	2738	1840	411	287
128bit	3081	2263	457	311
144bit	3473	2781	506	338
160bit	3900	3299	558	368
176bit	4318	3917	612	399
192bit	4843	4503	670	431
208bit	5436	5137	733	464

Table 2 shows the experimental data of the performance tests on RC6 and RC6-plus encryption algorithms for text and numeric datasets, respectively, and Fig. 4 shows the visual line graph corresponding to Table 2. The data are recorded in the table.

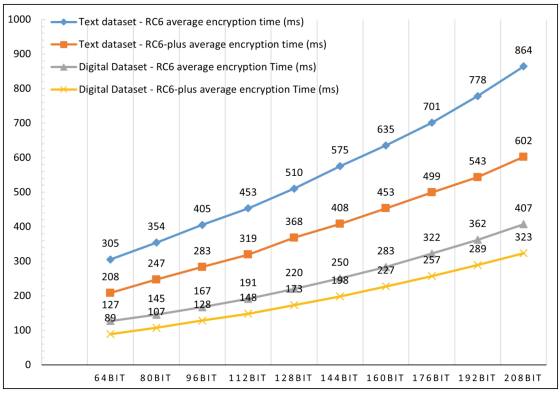

Fig. 4. Visual line chart of performance comparison of encryption algorithms for text and digital datasets

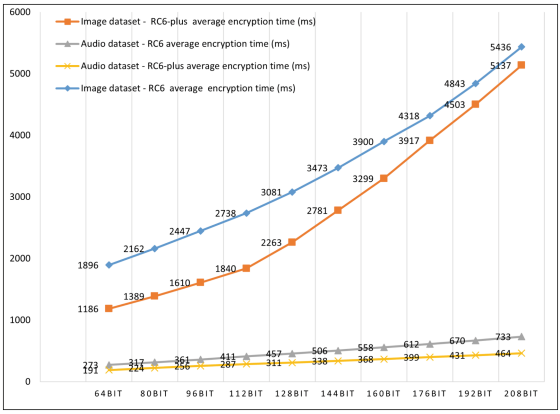

Fig. 5. Performance comparison of image and audio dataset encryption algorithms

When the Key length is equal to 64bit, the Text dataset RC6-plus average encryption time is equal to 305 ms, while the Text dataset RC6-plus average encryption time is only 208 ms, RC6-plus encryption time is less than RC6; When the Key length increases to 208 bits, the advantage of RC6-plus is more apparent, and the average encryption time of RC6-plus is 602 ms, which is 262 ms less than that of RC6.

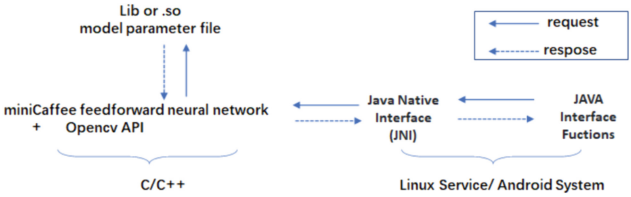

Fig. 6. AI-IOT software decoupling architecture

When the Key length equals 64 bits, the Digital dataset RC6 average encryption time equals 127 ms, while the Digital dataset - RC6-plus average encryption time only takes 89 ms; when the Key length increases to 208 bits, the Digital dataset RC6 average encryption time equals 407 ms, while the Digital dataset - RC6-plus average encryption time only takes 323 ms. When the Key length increases to 208 bits, the Digital dataset RC6 average encryption time equals 407 ms, while the Digital dataset - RC6-plus average encryption time is only 323 ms.

When the key length is 64 bits, the average encryption time of an image data set using the RC6 algorithm is 1896 milliseconds, while the average encryption time using the RC6-plus algorithm is only 1186 milliseconds, and the encryption time of the RC6-plus algorithm is significantly less than that of the RC6 algorithm; when the key length is increased to 208 bits, the advantage of RC6-plus algorithm is more prominent, and its average encryption time is 5137 ms, which takes 299 ms less than the RC6 algorithm. For audio data sets, the average encryption time of the RC6-plus algorithm is only 191 milliseconds when the key length is 64 bits. For digital data sets, the average encryption time of the RC6-plus algorithm is only 464 milliseconds when the key length is 208 bits.

From the results of the analysis of the above experimental data, we draw several conclusions:

1. The encryption time increases as the value of the Key length increases;
2. The encryption time will vary depending on the complexity of the data;
3. The RC6-plus algorithm can be used on text datasets, digital datasets, image datasets, or audio datasets. The encryption performance of the algorithms on the audio data set is significantly better than RC6.
4. These data results show that the RC6-plus algorithm has a shorter encryption time than the RC6 algorithm for different data sets and critical lengths. The advantage is undeniable for longer key lengths.

The experimental data of encryption strength and encryption time of the RC6-plus algorithm do not include the experimental data of decryption of the RC6-plus algorithm. In practical applications, the decryption time and strength are usually related to factors such as the length of the key used for encryption, the data set, and the algorithm version. Usually, the decryption process of RC6-plus is similar to the encryption process, but the order of the keys is reversed. Therefore, the same key and algorithm parameters should be used in the decryption phase as in the encryption phase. In the decryption process, the

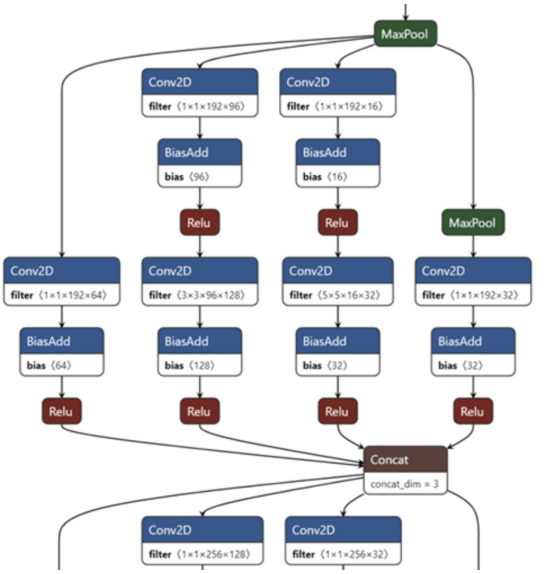

Fig. 7. Structure diagram of AI model before encryption (partial display)

RC6-plus algorithm will use the same algorithmic process but execute the algorithmic steps opposite to the encryption direction. The wheel keys are applied opposite to recover the original message. Therefore, the decryption process of the RC6-plus algorithm takes the same amount of time as the encryption process.

It is important to note that the key length and algorithm parameters used in the RC6-plus encryption and decryption process must be the same to perform the decryption correctly. If a different key length or parameters are used in the decryption phase than in the encryption phase, the decryption process may fail or get an incorrect message.

In addition, this paper also designs the AI-IoT software decoupling architecture so that the API interface layer is decoupled from the JNI layer. The user can only see the outer interface call function but does not know the principle of the internal algorithm [7,9,16], as shown in Fig. 6. The forward inference neural network of the AI model is rewritten using the miniCaffe C++ language, and the source code is obfuscated so that even standard model visualization cracking tools cannot read or write to the model. With the above source code encryption process, the project source code can be compiled to generate .so files, making the system more private, as shown in Fig. 7 and Fig. 8.

Fig. 8. Data of the encrypted AI model file, garbled and unreadable (partial display)

5 Conclusions

In this paper, a model privacy protection solution is proposed to help protect the privacy of model owners and data owners in complex IoT application environments. The improved RC6-plus, JNI middleware approach can more effectively protect model privacy in the IoT environment. Experimental results demonstrate the reasonableness and effectiveness of the approach. Considering AI model applications are increasingly used in new smart IoT software, hardware application products, and other services, and the arithmetic performance of cloud-based servers far exceeds the hardware configuration on the IoT side, this research can be further extended to model encryption and decryption of cloud-based server applications and data privacy protection during remote data transmission of IoT.

References

1. Ahmed, H.E.D.H., Kalash, H.M., Allah, O.F.: Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images. In: 2007 International Conference on Electrical Engineering, pp. 1–7. IEEE (2007)
2. Alenezi, M.N., Alabdulrazzaq, H., Mohammad, N.Q.: Symmetric encryption algorithms: review and evaluation study. Int. J. Commun. Netw. Inf. Secur. **12**(2), 256–272 (2020)
3. Ashraf, A.M., Elmedany, W.M., Sharif, M.S.: Secure IoT data transmission at physical layer using rc6 encryption technique. In: 2022 9th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 307–315. IEEE (2022)
4. Baños-Gonzalez, V., Afaqui, M.S., Lopez-Aguilera, E., Garcia-Villegas, E.: IEEE 802.11 ah: a technology to face the IoT challenge. Sensors **16**(11), 1960 (2016)
5. Dong, B., Shi, Q., Yang, Y., Wen, F., Zhang, Z., Lee, C.: Technology evolution from self-powered sensors to AIoT enabled smart homes. Nano Energy **79**, 105414 (2021)
6. Fang, K., Yang, B., Zhu, H., Lin, Z., Wang, Z.: Two-way reliable forwarding strategy of RIS symbiotic communications for vehicular named data networks. IEEE Internet Things J. (2022)

7. Faragallah, O.S., et al.: Efficiently encrypting color images with few details based on RC6 and different operation modes for cybersecurity applications. IEEE Access **8**, 103200–103218 (2020)
8. Farooq, M.U., Waseem, M., Mazhar, S., Khairi, A., Kamal, T.: A review on internet of things (IoT). Int. J. Comput. Appl. **113**(1), 1–7 (2015)
9. Ghadirli, H.M., Nodehi, A., Enayatifar, R.: An overview of encryption algorithms in color images. Signal Process. **164**, 163–185 (2019)
10. Gokhale, P., Bhat, O., Bhat, S.: Introduction to IoT. Int. Adv. Res. J. Sci. Eng. Technol. **5**(1), 41–44 (2018)
11. Grichi, M., Abidi, M., Jaafar, F., Eghan, E.E., Adams, B.: On the impact of interlanguage dependencies in multilanguage systems empirical case study on java native interface applications (JNI). IEEE Trans. Reliab. **70**(1), 428–440 (2020)
12. Hwang, S., Lee, S., Kim, J., Ryu, S.: Justgen: effective test generation for unspecified JNI behaviors on JVMs. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1708–1718. IEEE (2021)
13. Jihui, Y., Qinian, Z., Zhenhao, Z.: Cloud storage and security technology based on the the internet of things. ZTE Technol. J. **18**(6), 12–16 (2012)
14. Lee, S., Lee, H., Ryu, S.: Broadening horizons of multilingual static analysis: semantic summary extraction from c code for JNI program analysis. In: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, pp. 127–137 (2020)
15. Li, B., Feng, Y., Xiong, Z., Yang, W., Liu, G.: Research on AI security enhanced encryption algorithm of autonomous IoT systems. Inf. Sci. **575**, 379–398 (2021)
16. Liu, F., Koenig, H.: A survey of video encryption algorithms. Comput. Secur. **29**(1), 3–15 (2010)
17. Lu, H., Jin, C., Helu, X., Zhu, C., Guizani, N., Tian, Z.: AutoD: intelligent blockchain application unpacking based on JNI layer deception call. IEEE Netw. **35**(2), 215–221 (2020)
18. Manikandan, V., Amirtharajan, R.: On dual encryption with RC6 and combined logistic tent map for grayscale and DICOM. Multimed. Tools Appl. **80**(15), 23511–23540 (2021)
19. Ming, H., Jun, J., Xiaohu, C., Guohua, C.: Technology and security of internet of things. Comput. Secur. **4**, 49–52 (2011)
20. Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., Shenker, S.: Software-defined internet architecture: decoupling architecture from infrastructure. In: Proceedings of the 11th ACM Workshop on Hot Topics in Networks, pp. 43–48 (2012)
21. Ren, W., et al.: Privacy-preserving using homomorphic encryption in mobile IoT systems. Comput. Commun. **165**, 105–111 (2021)
22. Sajjad, M., et al.: Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. Futur. Gener. Comput. Syst. **108**, 995–1007 (2020)
23. Wang, Y., Liang, X., Hei, X., Ji, W., Zhu, L.: Deep learning data privacy protection based on homomorphic encryption in AIoT. Mob. Inf. Syst. **2021**, 1–11 (2021)
24. Xiaoqiang, Z., Mengmeng, W., Guiliang, Z.: Research on the new development of image encryption algorithms. Comput. Eng. Sci. **34**(5), 1–6 (2012)
25. Xu, J., Zhao, B., Wu, Z.: Research on color image encryption algorithm based on bit-plane and chen chaotic system. Entropy **24**(2), 186 (2022)
26. Zhu, H., Peng, Y., Xu, H., Tong, F., Jiang, X.Q., Mirza, M.M.: Secrecy enhancement for SSK-based communications in wireless sensing systems. IEEE Sens. J. **22**(18), 18192–18201 (2022)