Cryptographic Fingerprinting for Network Devices Based on Triplet Network and Fuzzy Extractors

Longjiang Li[1(✉)], Yajie Kang[1], Yukun Liang[1], Xutong Liu[1], and Yonggang Li[2]

[1] SICE, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China
longjiangli@uestc.edu.cn

[2] School of Information and Communication Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
lyg@cqupt.edu.cn
https://faculty.uestc.edu.cn/lilongjiang/

Abstract. Device fingerprinting is a key technology in cybersecurity, which enables organizations to identify potential vulnerabilities, gain valuable insights into their network infrastructure, and enhance overall defense mechanisms. However, the complexity and dynamics of cyberspace make it extremely challenging to generate unique, robust, and tamper-resistant device fingerprints. In this paper, we propose a cryptographic device fingerprinting framework for network devices, which utilizes triplet network to cluster the feature information of data samples into embeddings, and then apply fuzzy extractors to generate a cryptographic fingerprint for each device based on the feature information in each cluster. In order to overcome the discontinuity of embeddings in Hamming space output by triples networks, which degrades the robustness of fingerprints, we use gray code to transform embeddings before applying fuzzy extractors. The experimental results show that the method proposed can obtain unique and robust fingerprint encoding for the same type of device in noisy environments, and supports incremental fingerprint encoding for newly added devices through a small number of sample learning. The experimental results show that the classification accuracy reaches 99.99%, and the histogram of generated fingerprints conform to the Gaussian distribution, which reflects the excellent cryptographic characteristics.

Keywords: Device fingerprinting · Fuzzy extractor · Hamming space · Gray code · Impersonation attack

Supported by the National Natural Science Foundation of China (No. 61273235) and National Key Research and Development Program of China (No. 2022YFC3005702).

F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 428–444, 2024.
https://doi.org/10.1007/978-3-031-67162-3_27

1 Introduction

With the rapid development of wireless technology and the widespread coverage of wireless communication infrastructure, wireless networks have become a necessity in people's daily lives. However, the continuous emergence of wireless device impersonation attacks poses great challenges to the security of wireless networks [7]. In WiFi networks, attackers can use user identity information obtained from communication content to disguise themselves as legitimate devices, and then launch counterfeit attacks to deceive wireless access points (APs), thereby illegally accessing the network or conducting further attacks, such as Distributed Denial of Service (DDoS) and vulnerability scanning. Enforcing identity credentials is a widely recognized and effective way to prevent malicious attackers from stealing or even damaging the network. However, credentials usually rely on authentication by authoritative third parties and can only be used to verify the legitimacy of the server's identity. Moreover, most works related to credentials rely on cryptographic mechanisms to protect the distribution of credentials and to resist phishing attacks, but their effectiveness in wireless network security comes at a higher computational complexity and cost. Complex encryption algorithms limit the deployment of lightweight devices due to computational power limitations. Especially, in the open or compromised networking environments, the continuous upgrading of existing and cracking technologies also poses a threat to its security.

Fortunately, a class of emerging methods known as device fingerprinting [19] provides another means of confirming the user's identity. The work of Radhakrishnan et al. [12] shows that even simple traffic characteristics, such as message arrival intervals, can reflect the uniqueness of a device. In the complex IoT application scenario, many studies attempted to recognize device identities based on passive observation of network traffic [11]. Basically, most of these methods model device fingerprinting as a classification problem, which can be handled by traditional classification methods such as decision trees and support vector machines(SVM), or Deep-Learning Neural Network(DNN) methods [9]. However, the device fingerprinting method only solves the device recognition problem, but does not solve the impersonation attack problem. Once the adversary actively steals or inadvertently caches the traffic characteristics of the device, it is possible to imitate the traffic characteristics of the device through traffic shaping. To the best of our knowledge, few references have taken into account both device fingerprinting and impersonation attacks.

In this paper, we propose a cryptographic device fingerprinting framework, which can generate device fingerprints that are difficult to impersonate by combining triplet network [13,15] and modified fuzzy extractors. Triplet network is a deep learning model, which is able to learn useful representations by distance comparisons. Unlike traditional deep learning networks, such as convolutional neural networks (CNN), triplet networks are able to adapt more easily to data updates through incremental learning. Then, fuzzy extractors [4] are applied to construct a cryptographic unique key for each device by extracting stable signals from the output of triplet network and added noisy data. In order to overcome

the discontinuity of embeddings in Hamming space output by triples networks, which degrades the robustness of fingerprints, we extend the fuzzy extractors by applying gray code [6] to transform embeddings as inputs.

The main innovations and contributions of this paper include the following aspects:

- A framework for combining triplet network and modified fuzzy extractor for generating cryptographic device fingerprints.
- A method of smoothing embeddings in Hamming space, which uses gray code to convert the output of triplet networks, so that the fuzzy extractors have more robust performance.
- Through simulation and experiments, the results show that the device fingerprints generated by the proposed method have a good random distribution, which reflects the excellent cryptographic characteristics.

2 Background and Related Work

2.1 Traditional Network Device Identification

Many existing network monitoring technologies such as Censys, Shodan and ZoomEye, use IP and MAC information as the main means of device identification. However, the IP and MAC information is easily modified or forged at the software level, which makes the identity information of the device not very reliable.

2.2 Existing Identification Methods for Network Devices

In recent years, a particular technology called Physical Unclosable Function(PUF) has received widespread attention. PUF is a promising technique that utilizes inherent manufacturing differences between devices to achieve identity recognition. Suh et al. [14] utilized the variable delay characteristics of integrated circuit ICs to introduce PUF for device identity identification and verification. Afterwards, Wang et al. [17] proposed a solution of PUF to implement mutual authentication mechanism between sensors, in which a coordinator is needed as an assistance. Zhang et al. [20] proposed a simplified identity verification solution that can resist known attacks such as impersonation, replay, and tamper attacks, but some security requirements of anonymity, mutual authentication and non Linkage were not taken into account.

Although PUF is promising for the security requirements of device authentication without the need for complex encryption algorithms or expensive hardware conditions, the gap between promise and reality still exists. For instance, some work based on commercial chips shows that some machine learning attack methods are able to construct RFID simulators to achieve cloning of commercial PUF-based RFID chips at a rather low cost [1].

2.3 Device Fingerprinting Methods in Cyberspace

There are already many studies on designing device authentication credentials based on device hardware level information, such as [3,8,12]. Due to the slight differences in the hardware composition of the device during the factory process, and the different user environments have different degrees of influence on the equipment components after leaving the factory, which leads to drift of the local hardware clock of the device. Some experimental results show that the degree of clock drift can be described by drift rate, which can be used for distinguishing devices with some good characteristics, such as measurability, uniqueness, distinguishability and stability. Radhakrishnan et al. [12] proposed a technology called GTID, which collects network traffic and uses artificial neural network to create unique and replicable devices and device Type signature. Fang et al. [5] design a device identity authentication mechanism using device hardware features based on webpage loading. The rationality is that while different hardware devices may lead to the different throughput when loading the same webpage, so the time traces generated during the webpage loading process can be used as device fingerprint features. Similarly, based on network traffic, Bezawada et al. [2] use passive detection to train machine learning classifiers to identify equipment by capturing the session message of wireless devices in the network space and extracting the characteristics of the header and load, but only at the device type level can the recognition accuracy meet the requirements.

2.4 Cryptographic Key Generations

Most cryptographic methods are used to secure communications through encryption and decryption algorithms, rather than being used to identify devices or users. Almost all cryptographic key generation methods emphasize the randomness of the key. In general, cryptographic keys can be generated through either secret key exchange or physical-layer wiretap codes [10]. Typical key exchange methods include Diffie-Hellman algorithm (D-H), lattice-based cryptography, and quantum key distribution (QKD), while physical-layer wiretap methods usually are called physical-layer key generation, which constructs cryptographic keys by extracting randomness from fading channels.

In contrast, fuzzy extractors [4] is a transformation method that can convert repeated noisy readings of a secret into the same uniformly distributed key. Since the key is in a uniformly random distribution, it is hard to guess the repeated noisy readings if only these readings are not cached. This inspires us that fuzzy extractors can provide a promising method for resisting impersonation attacks.

3 Key Level Device Fingerprint Generation Method

3.1 Problem Description

For fingerprint generation methods, keys are commonly used to ensure the confidentiality of fingerprints, but symmetric encryption algorithms have a high risk

of being cracked, while asymmetric algorithms exchange memory consumption for the security of identity fingerprints. An effective and low-cost approach is to use a fuzzy extractor to generate physically non clonable fingerprints, making them tamper resistant.

Another challenge faced by fingerprint design based on device features is that the device feature information used needs to be distinguishable, and the more distinguishing the feature information from other devices, the less likely it is to be the same as other devices. However, another issue that needs to be considered is that the feature information needs to have no significant fluctuations within a limited time range. The selected feature information is expected to balance uniqueness and stability, but the fact is often difficult to achieve as desired. The feature information that is easy to collect in existing research is difficult to meet both requirements. Therefore, the key to fingerprint design schemes is to design appropriate methods to extract and process the collected feature information to meet the above two requirements.

It is worth mentioning that in terms of stability, in addition to considering that device fingerprints should maintain stability on a time scale, it is also important to consider that with the introduction of new devices in the network, the fingerprint generation of existing devices will not be affected by the new devices. The deep neural network model involved in the fingerprint generation process needs to be continuously retrained to adapt to changes in feature data, which is called incremental learning (IL). The traditional approach uses regular retraining to update the network, but its drawbacks are also evideideant, as the continuous addition of new devices can lead to a sharp increase in memory consumption and training time.

3.2 Basic Idea

The goal of this paper is to seek a fingerprint generation scheme that is based on the relatively unique and stable feature information of the device, and performs appropriate processing to generate fingerprints with immutability, uniqueness, and stability. At the same time, the likelihood of being impersonated should be as small as possible. The designed fingerprint generation mechanism ensures security while calmly responding to the scenario of constantly adding new devices to the system, and can train the network to recognize new devices with almost no changes to the original model parameters.

Based on existing research, this paper selects the inter arrival time (IAT) between network device packets as the feature information used for device fingerprint generation. The main consideration is that IAT, as a hardware level feature of the device, has the ability to distinguish different devices and does not fluctuate significantly in a short period of time. For the IAT information collected over a period of time, this paper analyzes the statistical characteristics of the data and transforms them into a histogram array as input for the next step.

As shown in the Fig. 1, we use triplet network to perform two-dimensional spatial mapping on the processed IAT data histogram array, during which device inter class segmentation and intra class aggregation are achieved. The output of

triplet network is a multidimensional tensor, called an embedding, which consists a series of float numbers. If these floating-point numbers are directly represented as binary format, they are discontinuous in Hamming space. For example, the Hamming distance between "011" and "100" is 3, even though they differ by only 1 as integer numbers. In order to overcome the discontinuity of embeddings in Hamming space, we use gray code transformation to process these embeddings, as a gray code transformation enables that adjacent numbers have a single digit differing by 1. Finally, the two-dimensional encoding of IAT data is inputted into the fuzzy extractor to generate a unique and robust fingerprint encoding for the device.

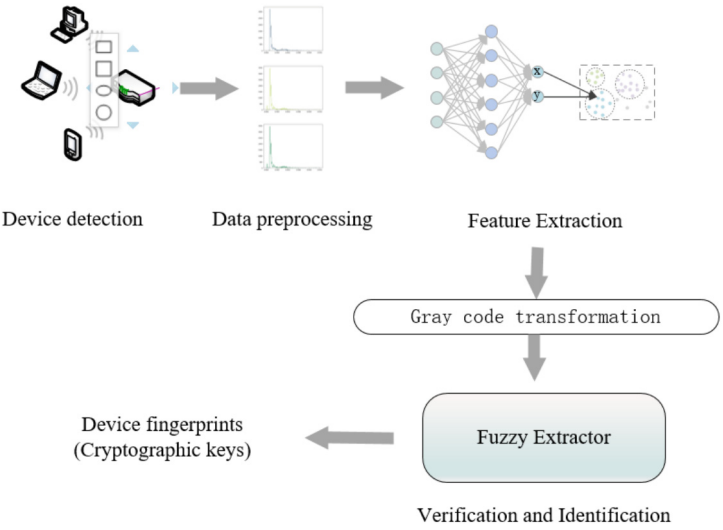


Fig. 1. Fingerprint Design Framework

3.3 Data Sample Collection

The performance and capability of device hardware can be reflected in the speed of sending and receiving data packets. Taking the packet sending process as an example, the hardware involved in the packet creation process includes CPU, L1/L2 cache, physical memory, DMA controller, Front-side bus, back-end bus, PCI bus and NIC. Therefore, devices composed of different hardware components will reflect the differences in device identity in the information of packet arrival delay.

The use of different device hardware components and usage losses will cause changes in the device clock offset. We collect the IAT in the network traffic information. In this study, we captured data packets of various types of wireless devices in isolated and open environments, and obtained IAT features of different devices based on timestamp information. Placing the tested equipment in an isolated environment in a shielding device can reduce the interference of the surrounding environment in achieving radio frequency. The campus network, as a

typical practical application scenario, is used for testing. The experimental process may be affected by interference from other frequency devices around it, but this can verify the effectiveness of this scheme in real application environments with noise interference.

When preprocessing IAT data, we focus on the statistical characteristics of the obtained IAT information over a period of time. Therefore, the original data is processed in the form of a histogram at a threshold of 0.01, which intuitively represents the distribution characteristics of device IAT information in various time intervals.

3.4 Generations of Feature Embeddings

In order to enhance the recognition ability of the feature encoder, we propose to use triple loss function [18] when training the feature extraction network. Through the comparison between samples (within class/between classes), the model has the ability to distinguish the similarity between samples. Using the feature code that has been aggregated within class and dispersed between classes to replace the feature tag to represent the feature set of the device can ensure that the constructed feature code meets the local sensitive hash characteristics.

The encoding of features in two-dimensional space is represented by $f(x)$. This article aims to ensure that feature x_i^n (anchor) of a device target is closer to all other features x_i^p (similar samples) of the same target, rather than any feature x_i^n (dissimilar samples) belonging to different devices. Therefore, the Loss function to be minimized is:

$$TL = \sum_{1}^{N} \left[\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 - \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 \right] \tag{1}$$

In this paper, we mainly use convolutional neural network (CNN, one-dimensional or two-dimensional) as the feedforward network. The ultimate goal of the ternary loss function is to shorten the distance between the reference sample and the same type of sample as far as possible, and at the same time distance between the reference sample and the different type of sample. These three feedforward networks share parameters. By using the same ternary loss function for training, we can learn the similarity of the same type of sample and the difference of the different type of sample at the same time. As shown in the Fig. 2, the iterative model is divided into two stages: training and testing. The first stage is a pre training process, which is based on the training dataset to train the feedforward network model with locally sensitive hash encoding ability, that is, the feature codes output by different types of samples have a longer distance, and the feature codes output by similar samples have a closer distance. The recognition process of cyberspace information is the second stage, with the goal of enabling iterative models to recognize online devices. The specific approach is to collect IAT data samples of online devices and enhance the iterative network based on the ternary loss function. The preprocessed test data samples

can be trained using a feedforward network to generate two-dimensional feature codes for each test sample, which can have locally sensitive hash characteristics.

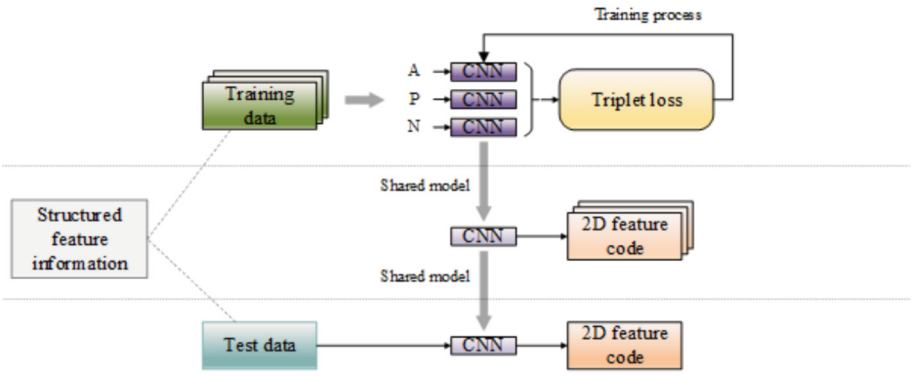


Fig. 2. Triplet network process

3.5 Gray Code Transformation

Gray code transformation is used to overcome the discontinuity of embeddings in Hamming space, because a gray code has a perfect characteristics that adjacent numbers have a single digit differing by 1.

The pseudocode of gray code transformation is shown in Algorithm 1. Since an embedding is array of integer numbers. $BinaryRepresent()$ is a function that converts an integer number to a vector of binary bits. Correspondingly, $IntegerRepresent()$ is a function that converts a vector of binary bits to an integer number. After gray code transformation is applied, embeddings have been converted into Hamming space, ready for being further processed by fuzzy extractors.

3.6 Fingerprint Generation

As Radhakrishnan et al. [12] pointed out that each device has non-replicable uniqueness in its traffic pattern, which can be seen as a random source. It is believed that even if there are differences in the characteristics exhibited by the device at different times, analyzing the feature data of the same device at all times can reveal a pattern where stable features uniquely correspond to the device identity.

Dodis et al. [4] proves that the information reconciliation step is able to eliminate of noise from the actual traffic data collected from network device. The function structure of fuzzy extractor used in this paper is shown in Fig. 3, where BCH encoding is used to perform the function of information reconciliation and SHA 256 is a 256-bit cryptographic hash function.

Data:
$x = [x_1, x_2, ..., x_w]$ denoting embeddings in the format of multi-dimensional tensor ;
Result: $y = [y_1, y_2, ..., y_w]$ denoting embeddings in the format of gray code array;
initialization;
for *i = 1: w* **do**
  $s \leftarrow BinaryRepresent(x_i)$;
  $gray \leftarrow 0$;
  $gray(1) \leftarrow s(1)$;
  for *i = 2: s(2)* **do**
    $gray(i) \leftarrow XOR(x_{i-1}, x_i)$;
  end
  $y_i \leftarrow IntegerRepresent(gray)$;
end
return y;

Algorithm 1: The pseudocode of gray code transformation

In our implementation, the output of triplet network is a two-dimensional embedding. In order to overcome the discontinuity of embeddings in Hamming space, gray code transformation is applied to encode the two-dimensional embeddings. Figure 4 illustrates the process that gray code transformation converts a two-dimensional embedding to the Hamming space. Correspondingly, the distance of the anchor point in the middle of the two-dimensional plane will be converted into the Hamming distance between binary codes. The feature information from the same device has a smaller Hamming distance after being processed and encoded, so it will have a greater chance to be determined as the same device when passing through the fuzzy extractor, i.e., gray code transformation can contribute to improving the accuracy of fingerprint generation. For the characteristic information of different devices, the processing of triplet network and gray code coding can also increase the Hamming distance after coding, and reduce the occurrence of misjudgments.

4 Device Fingerprint Update

4.1 Requirements of Update

Almost all device fingerprinting methods rely on classification algorithms, such as decision tree and neural networks, to identify device identities or types. However, due to the instability of the network, adding devices to the original system means retraining the model. Abandoning historical data may lead to significant differences in fingerprints of the same device before and after the new model is generated, which may lead to potential catastrophic forgetting problems. Therefore, when implementing the fingerprint generation and verification system, it is also extremely necessary to design an effective incremental training method to

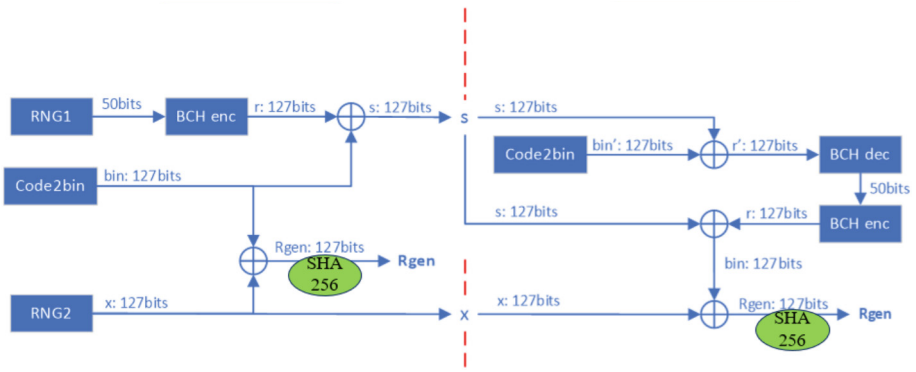


Fig. 3. The basic function structure of fuzzy extractors

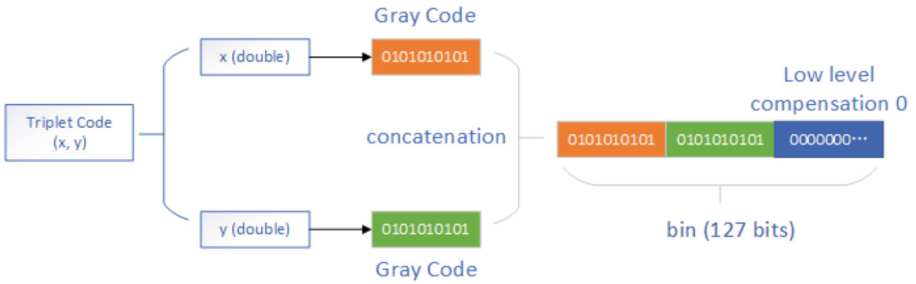


Fig. 4. Gray encoding processing

maintain the stability of fingerprint generation, improve the efficiency of model training, and reduce memory consumption.

Although the type recognition algorithms based on deep learning have strong feature extraction and representation capabilities, traditional deep learning networks such as Convolutional neural network (CNN) need to rely on a large number of high-quality training data. Due to the complexity and dynamics of the cyberspace, it is often difficult to obtain a large amount of high-quality training data, especially when new devices are added into the network. At this time, it is necessary to update the network model in a timely manner to identify new devices. Therefore, we employ an iterative learning method based on ternary loss function to incrementally train the neural network.

4.2 Incremental Training Process

The working process of the iterative learning method can be divided into two stages: offline and online.

The offline stage is a pre-training process based on the training dataset, which enables the feedforward network model to have local sensitive hash encoding

ability. The expected training result is that the distance between the feature codes output by different types of samples is longer, and the distance between the feature codes output by similar samples is closer.

The process of identifying cyberspace information, known as the online stage, aims to equip the iterative model with the ability to identify online devices. The specific approach is to collect M data samples from online devices, mix them with a certain number of old data samples (which should contain data from other devices), construct triplets, and then conduct incremental training on the iterative network based on the ternary loss function.

As shown in Fig. 5, at the beginning of training, fully load the pre-trained model parameters as the initial parameters of the model. At the end of the training, the new model parameters obtained from incremental training are fully saved as the initial parameters for the next incremental training. At the same time, classification algorithms are used to establish the mapping relationship between feature codes and labels. After collecting test data samples, the trained feedforward network is used to generate feature codes for each test sample, and then classification algorithms are used to achieve target classification and recognition. The iterative model based on the ternary loss function can achieve high recognition accuracy through incremental training, i.e., the original model parameters are kept and only a small number of new class samples are needed for training.

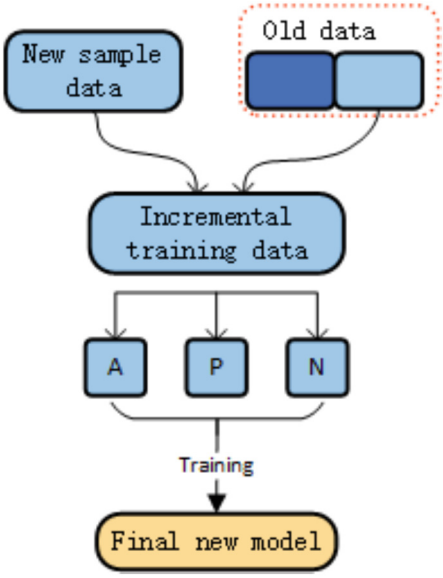


Fig. 5. Incremental training process

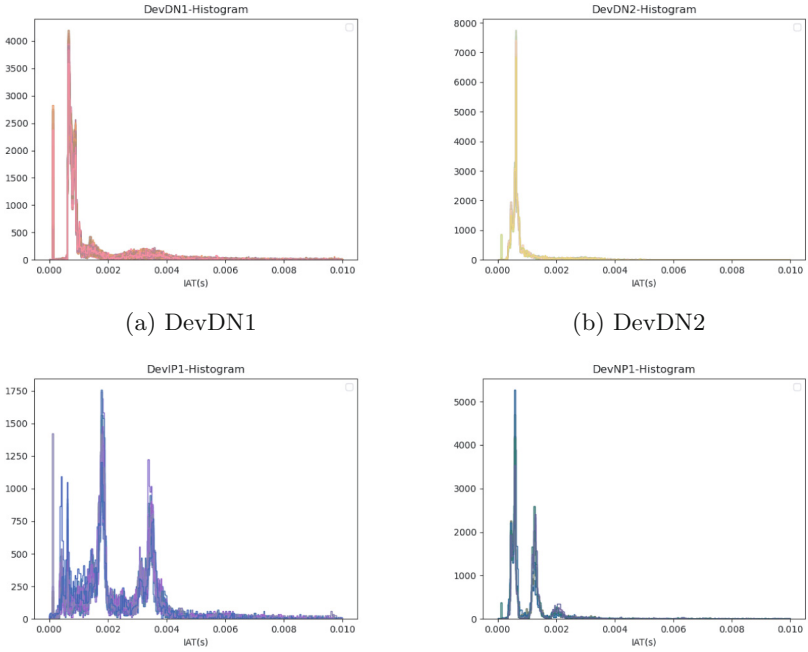
5 Experimental Implementation and Result Analysis

In this section, we evaluate the performance of the proposed framework from three aspects in an isolated network environment, including analyzing the uniqueness, robustness, and tamper resistance of each stage of device fingerprint generation.

The dataset we used in our experiment are GTID data collected by the CRAWDAD team in 2014 [16]. They captured over 300GB of traffic data in isolated experimental environments and campus network environments, respectively. The isolated experimental environment can shield the impact of wireless devices other than experimental devices on the experiment, while the campus network environment verifies the performance of the proposed scheme in practical application scenarios.

5.1 Data Preparation

Taking the device traffic data captured in an isolated network environment as an example, a histogram is used to demonstrate the device differentiation ability of IAT data. As shown in Fig. 6, there are significant differences in the distribution of IAT data among different types of devices shown in Fig. 6b, Fig. 6c and Fig. 6d.



(a) DevDN1 (b) DevDN2

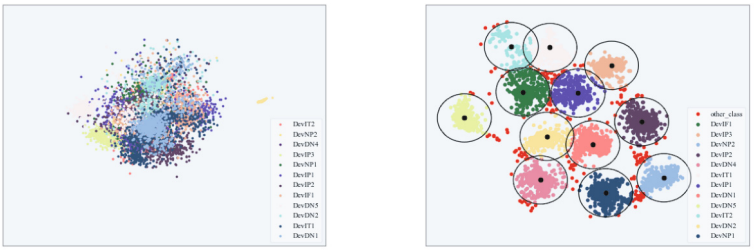
(c) DevIP1 (d) DevNP1

Fig. 6. IAT distribution of different devices

Although the trends of the same type of device are relatively consistent, the distribution still varies in Fig. 6a and Fig. 6b. These experimental data provides a support for the reasonability of adopting IAT data for device fingerprinting.

5.2 Numerical Results

Verification of Fingerprinting Capability. Since the packet arrival interval has been proved to be significantly related to the device identity, the triple loss function can be used to effectively distinguish the corresponding relationship between the feature tag and the device target in each type of feature. This network has been trained so that the square of the distance in multi-dimensional space directly corresponds to the similarity between feature labels, that is, features from the same device have smaller distances, while features from different devices have larger distances. Taking IAT data of multiple types of wireless devices in an isolated environment as an example, the multi-dimensional distribution effect of different devices through randomly initialized networks and Triplet networks is shown in Fig. 7.

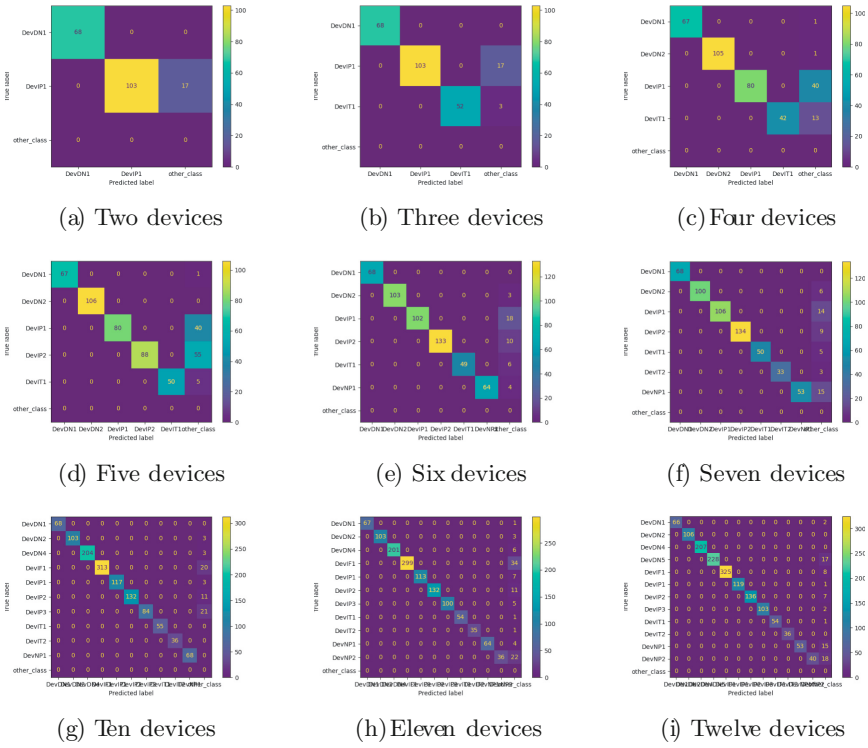


(a)Initial two-dimensional distribution of the dataset

(b) Two-dimensional distribution after training of ternary loss function

Fig. 7. Two-dimensional distribution of different devices

Comparison of Accuracy. In isolated network environments, the classification accuracy of two-dimensional encoding processed by triplet networks under incremental training can reach over 90% in most cases, as shown in Fig. 8. It can be observed that if the new device is a type of device that has never existed in the existing training model, the overall classification accuracy of the new device will remain relatively unchanged or even improve. However, when the new device type already exists in the existing device, the classification accuracy will have a slight degradation trend. Among them, the experimental data of the new devices DevDN2 in Fig. 8c and DevIP2 in Fig. 8d show good performance, but the accuracy of device recognition for the existed devices decreased by 7.53% and 4.77%, respectively.



(a) Two devices (b) Three devices (c) Four devices

(d) Five devices (e) Six devices (f) Seven devices

(g) Ten devices (h) Eleven devices (i) Twelve devices

Fig. 8. Confusion matrix of two-dimensional coded classification results

Effectiveness of Gray Code Transformation. Figure 9 shows the effectiveness of fuzzy extractors and gray code transformation on prediction accuracy, respectively. It can be seen that fuzzy processors can improve the accuracy by 6% if gray code transformation is not applied. After gray code transformation is activated, the accuracy has an increase by 0.4%. And in the process of incremental training, the accuracy fluctuates to a relatively large extent along with the growth of the number of devices. In contrast, with fuzzy extractors and gray code transformation, the curve becomes relatively stable, basically above 99.8% accuracy. This is reasonable because the fuzzy extractor has ability to reduce the noise in feature data, and gray code transformation is able to reduce discontinuities of embeddings in Hamming space.

From perspective of tamper resistance, the fuzzy extractors enable that only the output fingerprint should be retained, while the device's traffic data can be discarded. Since SHA256 used in fuzzy extractors is a one-way hash function, it is basically impossible to obtain useful information about device. It cuts off the possibility of intruders obtaining device information through cracking message data to disguise themselves.

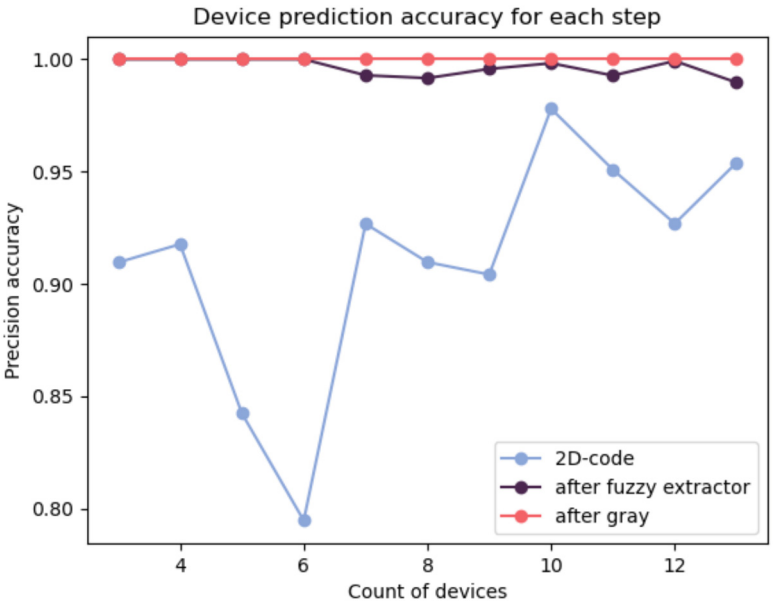
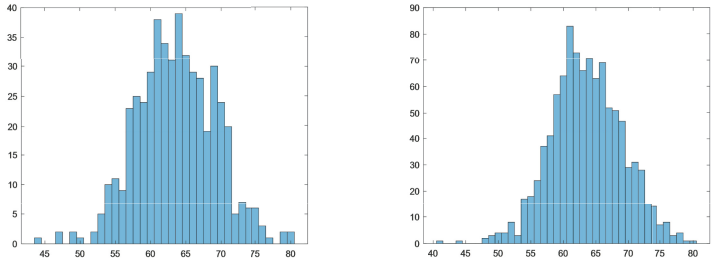


Fig. 9. Prediction accuracy in isolated network environments



(a) Key distribution for 500 devices

(b) Key distribution for 1000 devices

Fig. 10. Key distribution of fingerprints for 500 and 1000 devices

Key Distribution of Device Fingerprinting. Figure 10 shows the relationship between the histogram of key distribution of device fingerprinting and the number of devices. Since device fingerprints adopt 127-bit encoding, the histogram consists of 128 bins. The value of each bin is set to the number of 1 in the binary representation of each embedding, so its range is from 0 to 127, totally 128 bins. It can be seen that the histogram in Fig. 10b is slightly closer to the Gaussian distribution that in Fig. 10a. This implies that the generated device fingerprint carries very little information about network traffic feature,

which greatly increases the difficulty of impersonation attacks and reflects the excellent cryptographic characteristics.

6 Conclusion

This article proposes a cryptographic device fingerprinting framework, which combines triplet network and fuzzy extractor to generate fingerprints for network devices. Triplet networks are employed to extract device characteristics with saliency from network traffic and express them as multi-dimensional embeddings, while fuzzy extractors are committed to generating unique, robust, and tamper-resistant device fingerprints from the multi-dimensional embeddings. Moreover, in order to overcome the discontinuity of embeddings in Hamming space, gray code transformation is introduced to transform embeddings before applying fuzzy extractors. The experimental results validate the effectiveness of fingerprinting capability based on IAT data and demonstrate that the prediction accuracy is improved as gray code transformation significantly mitigates the discontinuity of embeddings in Hamming space. Moreover, the proposed framework enables excellent random distribution characteristics to the device fingerprints, while taking into account the prediction accuracy and anti-impersonation attacks.

Although both triplet networks and fuzzy extractors themselves have been widely studied and applied, the combination of the two has still received little attention. Basically, the idea in this paper is generic and may also be generalized to other applications such as extracting cryptographic keys from biological data. These issues deserve further in-depth study.

References

1. Becker, G.T.: The gap between promise and reality: on the insecurity of XOR arbiter PUFs. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 535–555. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48324-4_27
2. Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., Ray, I.: Behavioral fingerprinting of IoT devices. In: Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, pp. 41–50 (2018)
3. Cohen, M.I.: Source attribution for network address translated forensic captures. Digit. Investig. **5**(3–4), 138–145 (2009)
4. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_31
5. Fang, P., Huang, L., Xu, H., He, Q.: Smart device fingerprinting based on webpage loading. In: Chellappan, S., Cheng, W., Li, W. (eds.) WASA 2018. LNCS, vol. 10874, pp. 127–139. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94268-1_11
6. Grover, L.: Weighted code approach to generate gray code. IEEE Potentials **34**, 39–40 (2015). https://doi.org/10.1109/MPOT.2013.2295874

7. Hamdaoui, B., Elmaghbub, A.: Deep-learning-based device fingerprinting for increased LoRa-IoT security: sensitivity to network deployment changes. IEEE Netw. **36**, 204–210 (2022). https://doi.org/10.1109/MNET.001.2100553
8. Lanze, F., Panchenko, A., Braatz, B., Zinnen, A.: Clock skew based remote device fingerprinting demystified. In: 2012 IEEE Global Communications Conference (GLOBECOM), pp. 813–819. IEEE (2012)
9. Liu, Y., Wang, J., Li, J., Niu, S., Song, H.: Machine learning for the detection and identification of internet of things devices: a survey. IEEE Internet Things J. **9**(1), 298–320 (2021)
10. Mathur, S., Trappe, W., Mandayam, N., Ye, C., Reznik, A.: Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom 2008, pp. 128–139. Association for Computing Machinery, New York (2008). https://doi.org/10.1145/1409944.1409960
11. Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.R., Tarkoma, S.: IoT sentinel: automated device-type identification for security enforcement in IoT. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2177–2184. IEEE (2017)
12. Radhakrishnan, S.V., Uluagac, A.S., Beyah, R.: GTID: a technique for physical device and device type fingerprinting. IEEE Trans. Dependable Secure Comput. **12**, 519–532 (2015). https://doi.org/10.1109/TDSC.2014.2369033
13. Sirinam, P., Mathews, N., Rahman, M.S., Wright, M.: Triplet fingerprinting: more practical and portable website fingerprinting with n-shot learning. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, pp. 1131–1148. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3319535.3354217
14. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of the 44th Annual Design Automation Conference, pp. 9–14 (2007)
15. Thammasorn, P., et al.: Nearest neighbor-based strategy to optimize multi-view triplet network for classification of small-sample medical imaging data. IEEE Trans. Neural Netw. Learn. Syst. **34**, 586–600 (2023). https://doi.org/10.1109/TNNLS.2021.3059635
16. Uluagac, A.S.: Crawdad gatech/fingerprinting (2022). https://doi.org/10.15783/C78G67, https://dx.doi.org/10.15783/C78G67
17. Wang, W., Shi, X., Qin, T.: Encryption-free authentication and integrity protection in body area networks through physical unclonable functions. Smart Health **12**, 66–81 (2019)
18. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**(2) (2009)
19. Yang, K., Li, Q., Sun, L.: Towards automatic fingerprinting of IoT devices in the cyberspace. Comput. Netw. **148**, 318–327 (2019)
20. Zhang, W., Qin, T., Mekonen, M., Wang, W.: Wireless body area network identity authentication protocol based on physical unclonable function. In: 2018 International Conference on Sensor Networks and Signal Processing (SNSP), pp. 60–64. IEEE (2018)