Design of a Framework to Detect Device Spoofing Attacks Using Network Characteristics

Jaegwan Yu
Sungkyunkwan University

Hyoungshick Kim
Sungkyunkwan University

Eunsoo Kim
Sungkyunkwan University

Jun Ho Huh
Samsung Research

***Abstract*—This article proposes a generic framework to detect device spoofing attacks using physical network characteristics that are hard for an attacker to mimic, including received signal strength indicator and round trip time. A technological challenge with this approach is that those values can change over time and affect the detection accuracy. To overcome this challenge, we obtained the similarity of subsequent network behaviors by using a time series similarity measure. Our method continuously monitors physical network characteristics of a device, and looks for significant changes made in those monitored characteristics. Detected changes would indicate that a suspicious activity (e.g., device spoofing) has occurred. To demonstrate our implementation, we thoroughly tested the proposed framework on ZigBee (IEEE 802.15.4) wireless networks. We achieved a high F-measure accuracy of 0.96 when spoofing devices were located more than 5 m away from original devices.**

Wireless networks such as WiFi, Bluetooth, and ZigBee are popularly used as they can significantly improve accessibility, deployability, and usability of network devices. However, wireless network technologies could potentially increase the threat landscape and expose devices and users to more cyberattacks. For example, a seven-year-old child easily hacked into a public WiFi network after watching an online hacking tutorial.[1]

Digital Object Identifier 10.1109/MCE.2019.2953737
Date of current version 7 February 2020.

Published by the IEEE Consumer Electronics Society

Media access control (MAC) addresses are commonly used to control access in wireless networks (hereafter referred to as MAC-based access control), where only those devices that have previously authorized MAC addresses (i.e., MAC addresses are listed in a whitelist of known good addresses) can connect to a wireless network. However, MAC-based access control has several limitations. For example, maintaining an acceptable list of authorized MAC addresses (i.e., a whitelist of MAC addresses) is not only a cumbersome and challenging task for casual users, but tech-savvy users can also change MAC address to bypass this list quite easily.[2]

To address these limitations of MAC-based defense mechanism, several researchers proposed device fingerprinting methods using physical properties of wireless signals, which are usually hard to imitate. Faria and Cheriton[3] describe a method that uses received signal strength indicator (RSSI) to detect anomalous access points (APs). They consider RSSI values as a device specific feature that is hard to arbitrarily forge, and can accurately indicate the location of a device. There are similar studies focusing on using different statistical models, e.g., Gaussian mixture model,[4] K-means clustering,[5] and spatial correlation property of RSSI.[6] Most previous work has focused on developing a proper probabilistic model to better understand the actual characteristics of RSSI. However, such approaches often rely on having access to multiple network traffic monitoring devices to improve the overall reliability and accuracy. This requirement might not be acceptable in small network environments where only one AP is available.

This article presents a framework to detect device spoofing attack using the time series similarity measure between subsequent packet sequences. Unlike existing proposals (e.g., see the work done by Faria *et al.*)[3–6] that use fixed threshold values, our scheme runs on dynamically changing threshold values to deal with network conditions that change over time. To demonstrate the effectiveness of our scheme, we perform experiments with real world traffic dataset under various conditions, showing the F-measure accuracy of 0.96 when spoofed devices are located five or more meters away from original devices.

ATTACK SCENARIO

Wireless networking is inherently prone to cybersecurity threats. Unlike wired networks, wireless networks can be discovered by anyone nearby.

Due to such weaknesses in wireless networks, we assume that an attacker is capable of connecting a malicious device to victim's home devices (e.g., smart TV) by exploiting the victim's home network. This allows the attacker to play recorded movies, view pictures, or copy files from the victim's devices. The attacker may also try to take control of critical functions on various network-connected devices such as CCTV, door locks, and thermostats.[7] The data link layer has security solutions such as WEP and WPA, but these alone are not sufficient to properly control access to such devices. Additional security mechanisms are needed at the application layer to individually manage security policies for each device.

One possible technique is to perform access control based on identifiers that are unique to devices. The most widely adopted defense mechanism uses access control lists (ACL) created for the network addresses of known (authorized) devices. A network administrator defines security policies that allow only the known devices to connect with their IP and MAC addresses. MAC addresses are more frequently used since it is relatively easier to spoof IP addresses.

However, several limitations exist in the MAC-based access control mechanism. First, for typical users, it is quite cumbersome and challenging to timely update the ACL with fresh MAC addresses. If the coverage of the ACL is not sufficient, it can degrade the effectiveness of the ACL and discourage users from deploying it. Furthermore, the practical security of MAC-based ACL is weaker than expected because MAC addresses can be spoofed. "MAC spoofing" can be easily performed by a tool that experienced users can download and use. Ahmad[8] demonstrated that an attack device connected to the same wireless network can spoof and submit address resolution protocol replies to victim devices even if a

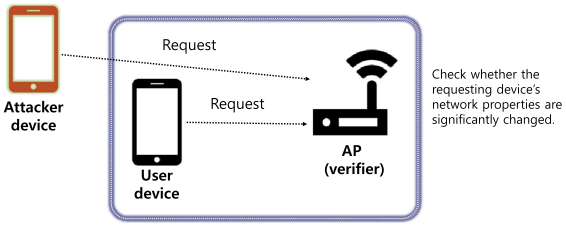


Figure 1. Device spoofing detection using the network characteristics of the user device.

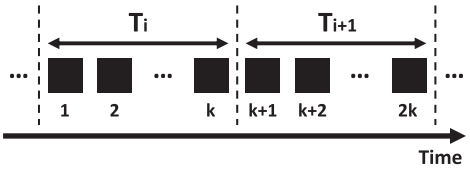


Figure 2. Network packets collected in two time windows.

well-accepted security standard, such as WPA pre-shared key, was used.

DEVICE SPOOFING DETECTION WITH NETWORK CHARACTERISTICS

The main idea is to monitor the physical network characteristics of a device, checking for any significant change in the network characteristics to detect messages delivered from spoofed devices. Our intuition is that the physical network characteristics of a device (e.g., RSSI) can change significantly depending on the physical device location, and be an effective feature for fingerprinting devices. Figure 1 shows how an attacker's spoofed device can be detected with our method. *"Verifier"* refers to a device that continuously monitors the changes (checking for anomalies) in the physical network characteristics of user devices. We imagine that APs could play the role of a *verifier*.

As one would imagine, a single network packet and its measurements are not sufficient to provide a reliable device fingerprint information. Therefore, we suggest using k network packets in a time window (see Figure 2). If the network characteristics of a device do not change significantly, packets in a time window T_i and packets in the next time window T_{i+1} are likely to show similar trends. Consequently, if the network characteristics of packets collected from a device during two time windows are significantly different and the device owner has not altered the geographical location of that device, there is some chance that a spoofing attack is being performed.

However, there are many technical challenges with this kind of approach that need to be addressed. For instance, one would have to investigate and analyze different network characteristics features that would be effective. One would have to optimize parameters like the window sizes and threshold values, and design algorithms to compare packets from two subsequent time intervals. The next sections discuss how we addressed those challenges through different experiments.

If the similarity score between packets in a time window T_i and packets in the next time window T_{i+1} is greater than a given threshold α, it can be classified as a suspicious (anomalous) device.

To implement this logic, we need to compute similarity scores between packets in consecutive time windows. We considered the following two popular time series algorithms: first, Euclidean distance, and second, dynamic time warping (DTW) distance (see the details in the work done by Yu *et al.*[9]).

CASE STUDY

To demonstrate the feasibility, we evaluate the performance of the proposed method on ZigBee networks, which are popularly used in Internet of Things applications. However, our technique can be generalized, and be applied to other wireless networking environments such as WiFi and Bluetooth.

For ZigBee, we consider the following three features that can be used for device fingerprinting. RSSI indicates the signal strength level of a received radio signal. The higher the RSSI value, the stronger the received signal. Round trip time (RTT) is the sum of propagation delays of a request packet and its response packet. RTT is generally affected by the geographical distance between two network parties involved in a communication. Link quality indicator (LQI) is a metric used to estimate the link quality of a received signal. It represents how easily a received signal can be demodulated by accumulating the magnitude of the error between ideal constellations and the received signal.

Procedure of Experiments

We designed experiments to evaluate the performance of the proposed method under various conditions. For simplification, we assume that a victim's device and an attacker's device are physically located 3 and m (either 4, 5, 6, 7, or 8) meters away from the *verifier*, respectively. That is, we considered two possible cases. The first case, referred to as a *normal* session, uses k consecutive packets from the victim's device. The second case, referred to as an *attack* session, uses k consecutive packets from the attacker's device.

The detailed experiment procedure is as follows. We fixed the position of the *verifier* device equipped with a ZigBee module, and placed the *prover* device (i.e., either victim or attacking device) with the same ZigBee module d meters away from the *verifier*. After positioning those devices, we started sending a 70-B packet from the *verifier* to the *prover*, observing RSSI, LQI, and RTT values for received packets. We repeated this procedure every 2000 times (every 1.5 s) for each of the *prover* device position. We collected 12 000 ZigBee packets in total. We performed this experiment in a university laboratory. In this laboratory setting, WiFi signals (sharing the same 2.4 GHz bandwidth range with ZigBee) generated from other wireless devices could have interfered with ZigBee communications, representing worst case environments. We wanted to show that the proposed method can achieve high detection accuracy even under such noisy environments.

We used two Probee Zu10 modules to measure RSSI, LQI, and RTT. RSSI and LQI were measured using the APIs provided in the ZigBee module. However, we computed RTT with our own implementation because RTT estimation was not supported.

Table 1. Relative weights of features for classification.

RSSI	RTT	LQI
0.574	**0.425**	0.001

We carefully analyzed the effects of each parameter, such as the window size, threshold, different features, and similarity algorithm, finding the optimal conditions to maximize detection accuracy.

Selection of Important Features

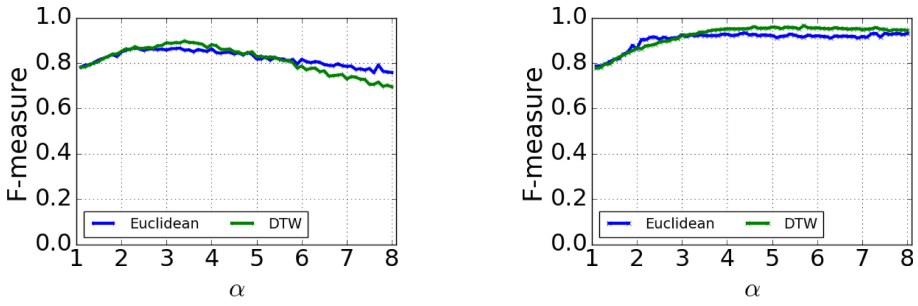
We used a tree-based feature selection algorithm implemented in the scikit-learn library to find an "optimal" subset of relevant features (RSSI, RTT, and LQI) for classification. Table 1 shows the relative weights of features for detecting spoofing attacks.

We can see that LQI is not significantly meaningful at all in designing a spoofing detection method for ZigBee networks. Therefore, we performed experiments using the two features (RSSI and RTT) only.

Results of Experiments

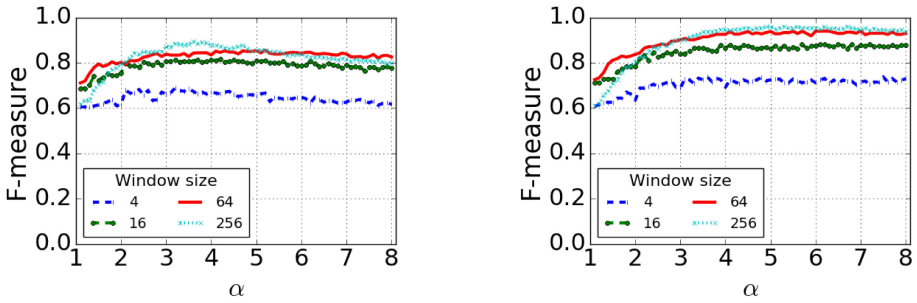
Effects of Similarity Measure Figure 3 shows the comparison between the F-measure obtained using Euclidean distance and DTW when the window size is 256.

In general, DTW produced more stable F-measure results compared with Euclidean distance and provides a wide range of choices for selecting α. DTW has a maximum F-measure (0.90 at 4 m and 0.97 at 5 m) than Euclidean



(a) D = 4m (b) D = 5m

Figure 3. F-measure results (Euclidean distance versus DTW) with varying α from 1.0 to 8.0. (a) D = 4 m. (b) D = 5 m.



(a) D = 4m (b) D = 5m

Figure 4. F-measure results for "RSSI only" with varying α from 1.0 to 8.0. (a) D = 4 m. (b) D = 5 m.

distance (0.87 at 4 m and 0.93 at 5 m). Interestingly, we can also observe an interesting pattern: DTW produced the better F-measure results with $\alpha \leq 5.0$ for $D = 4$ m (or with $\alpha \geq 3.0$ for $D = 5$ m).

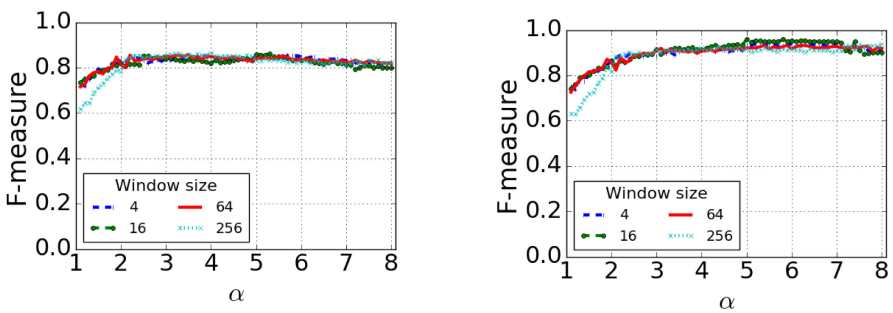
Based on those results, we recommend using DTW and present the results using DTW in the rest of this article.

Effects of Window Size To analyze the effects of window size on F-measure of the proposed method, we used four different window sizes (4, 16, 64, and 256). Here, window size indicates the number of packets to be used in a time interval. Therefore, a smaller window size is generally preferred to speed up the suspicious device detection process.

Figures 4 and 5 show the test results for "RSSI only" and "RSSI and RTT," respectively. We can see similar trends to those results except when the window size is 4. As mentioned above, for the windows size of 256, the proposed method produced the best results between $\alpha = 3.0$ and 4.0, when $D = 4$ m (or around $\alpha = 5.0$, when $D = 5$ m). However, the window size of 256 is relatively sensitive to α compared with the other window sizes. That is, if one wishes to use 256 as the window size, the threshold α should be selected carefully. The window size of 4 does not seem sufficient to achieve a high F-measure. Therefore, we recommend using at least 16 window size taking just 24 s.

Effects of α We analyzed the effects of α on the detection performance in more detail. Table 2 shows the classification accuracy results on precision, recall, and F-measure with α when the window size is 256. We can see that "RSSI only" and "RSSI and RTT" approaches generally produced highly accurate results.

Unsurprisingly, the test results were greatly changed with the physical distance between the victim device and the attacking device; when the attacking device is more distant, the proposed method could detect it with a higher accuracy. When that distance is 4 m, we can obtain the highest F-measure of 0.87 with either "RSSI only" or "RSSI and RTT" (at α = 3.0). When that distance is 5 m, the highest F-measure is 0.96 with either "RSSI only" or "RSSI and RTT" (at $\alpha = 5.0$). Therefore, our



(a) D = 4m (b) D = 5m

Figure 5. F-measure results for "RSSI and RTT" with varying α from 1.0 to 8.0. (a) D = 4 m. (b) D = 5 m.

Table 2. Classification accuracy with α (from 3.0 to 7.0), window size = 256 and DTW.

Distance	α	RSSI			RSSI+RTT		
		Precision	Recall	F-measure	Precision	Recall	F-measure
1 m	3.0	0.52	0.83	0.64	0.51	0.82	0.63
	5.0	0.53	0.93	0.67	0.52	0.93	0.67
	7.0	0.52	0.96	0.68	0.52	0.95	0.67
2 m	3.0	0.54	0.83	0.66	0.54	0.83	0.66
	5.0	0.54	0.93	0.68	0.53	0.92	0.68
	7.0	0.52	0.96	0.68	0.52	0.95	0.67
3 m	3.0	0.65	0.83	0.73	0.66	0.82	0.73
	5.0	0.60	0.93	0.73	0.60	0.95	0.74
	7.0	0.54	0.96	0.69	0.55	0.96	0.70
4 m	3.0	0.92	0.83	0.87	0.91	0.83	0.87
	5.0	0.80	0.92	0.86	0.81	0.92	0.86
	7.0	0.73	0.95	0.83	0.71	0.95	0.81
5 m	3.0	1.00	0.82	0.90	1.00	0.83	0.91
	5.0	0.99	0.93	0.96	0.99	0.93	0.96
	7.0	0.94	0.95	0.95	0.94	0.96	0.95

recommendation is to choose α between 3.0 and 5.0 for ZigBee networks. However, the proposed technique may not be effective when the physical distance between the victim device and the attacking device is less than 3 m because F-measure was less than 0.75 for such conditions.

CONCLUSION

We present a generic framework to detect device spoofing attacks with high accuracy. We demonstrated that suspicious network devices can be detected with their inherent network characteristics (e.g., RSSI and RTT) even when network address spoofing techniques are applied to mimic legitimate requests. We achieved a high F-measure accuracy of 0.96 using network characteristics when spoofing devices are located at more than 5 m away from original devices.

ACKNOWLEDGMENTS

This work was supported in part by the MSIP/IITP (No. 2016-0-00078) and in part by the ITRC program (IITP-2017-2015-0-00403).

REFERENCES

1. V. Woollaston, "Hacking Wi-Fi is child's play! 7-year-old shows how easy it is to break into a public network in less than 11 MINUTES," Mail Online, 2015.
2. W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," *IEEE Wireless Commun.*, vol. 9, no. 6, pp. 44–51, Dec. 2002.
3. D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using singalprints," in *Proc. 5th ACM Workshop Wireless Secur.*, 2006, pp. 43–52.
4. Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in *Proc. 27th Int. Conf. Comput. Commun.*, 2008, pp. 1768–1776.
5. Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, 2007, pp. 193–202.
6. P. Jokar, N. Arianpoo, and V. C. Leung, "Spoofing detection in IEEE 802.15. 4 networks based on received signal strength," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2648–2660, 2013.

7. A. Byrne, "CE devices hacked at black hat 2014 [Society News]," *IEEE Consum. Electron. Mag.*, vol. 4, no. 1, pp. 29–30, Jan. 2015.
8. M. S. Ahmad, "WPA Too!" in *Proc. DEFCON*, 2010, pp. 1–32.
9. J. Yu, E. Kim, H. Kim, and J. Huh, "A framework for detecting MAC and IP spoofing attacks with network characteristics," in *Proc. 2nd Int. Conf. Softw. Sec. Assurance*, 2016, pp. 49–53.

Jaegwan Yu is currently a Researcher with the Department of Cyber Warfare, LIGnex1, Yongin-si, South Korea. He received the MS degree from the Department of Computer Science and Engineering, Sungkyunkwan University, Seoul, South Korea. Contact him at jaegwan.yu@lignex1.com

Eunsoo Kim is currently working toward the Ph.D. degree at the Department of Computer Science and Engineering, Sungkyunkwan University, Seoul, South Korea. He received the M.S. degree from the Department of Computer Science and Engineering, Sungkyunkwan University. Contact him at eskim86@skku.edu

Hyoungshick Kim is currently an Assistant Professor with the Department of Software, Sungkyunkwan University, Seoul, South Korea. He received the Ph.D. degree from the Computer Laboratory, University of Cambridge, Cambridge, U.K. He is the corresponding author of this article. Contact him at hyoung@skku.edu

Jun Ho Huh is currently a Senior Cybersecurity Engineer with Samsung Research, Seoul, South Korea. He received the Ph.D. degree in cybersecurity and trustworthy computing from the University of Oxford, Oxford, U.K. His primary research interest is usable security, including usable authentication systems for mobile and IoT devices. Contact him at junho.huh@samsung.com