Rogue AP Detection using Similarity of Backbone Delay Fluctuation Histogram

Ziwei Zhang
Nagoya University
Nagoya, Japan
choshibi@net.itc.nagoya-u.ac.jp

Hirokazu Hasegawa
Nagoya University
Nagoya, Japan
hasegawa@icts.nagoya-u.ac.jp

Yukiko Yamaguchi
Nagoya University
Nagoya, Japan yamaguchi@itc.n
agoya-u.ac.jp

Hajime Shimada
Nagoya University
Nagoya, Japan
shimada@itc.nagoya-u.ac.jp

***Abstract*— Nowadays, wireless LAN service has been taken for granted by everyone. However, there is an increase in the occurrence of cyber threats to wireless LAN. For example, one attack called Evil-Twin Attack places a rogue access point (AP) with the same SSID as the legitimate one so clients connect to it unknowingly. Once attacked, all of the traffic moving across the network is sniffed by attackers. In this paper, we propose a method to detect the rogue AP by comparing different delay fluctuations in the backbone network. We define a delay in the backbone network as a subtraction result of the ICMP round trip time from client to first gateway or router and from client to the Internet server. To reduce deviation, we gathered 100 delay samples and created histogram-based vectors for a trial. We compared delay histogram vectors with cosine distances among six different wireless networks, which have different backbone networks. We confirmed that there is a large difference in cosine distance values between the same and different SSIDs on different days. We propose a method to detect the rogue AP, which compares a current delay histogram and a delay histogram from past days then compares the result with the threshold. The results showed that the proposed method can detect rogue APs with 86.67 percent accuracy under 6.25 percent false positive rate.**

Keywords— Evil Twin Attack, Rogue Wireless AP, Backbone N etwork Delay

I. INTRODUCTION

The Internet is becoming one of the most important infrastructures in society. In particular, wireless LAN or Wi-Fi is growing explosively because of its convenience and flexibility. Not only can you use it at home, but also when in the library, cafes or hotels. Besides this, devices including smartphones, laptops or even cameras can all be connected to wireless networks.

Meanwhile, cyber threats have to be considered more seriously due to the rising numbers of cyber-attacks. Since the connection is through the air, access to public Wi-Fi has various risks such as Denial of Service or being eavesdropped on through a man-in-the-middle attack [1]. Once the victim's device is connected to the malicious access point, all the traffic can be spied on no matter of what purposes the victims are using the Internet for, such as paying for online shopping by credit card, or uploading a confidential file to private websites that only their company employees can access. In addition, attackers can create a fake website and redirect the victims to it. In any event, important information will be obtained by the attackers.

There is a type of attack called the Evil Twin Attack (ETA), in which, attackers create a rogue access point (rogue AP) that hijacks the wireless connection from clients by providing the same SSID with a stronger signal than the legitimate AP. In addition, to convince the victims that the rogue AP is not a fake AP, in many cases attackers provides the Internet connection to clients. To provide such connectivity, attackers may route the traffic either through a legitimate AP (relay based rogue AP), or use different backbone networks (different network-based rogue AP) [2, 3].

Rogue AP-based attack is a big threat for wireless LAN service providers; thus, it is important to develop a method to detect rogue APs. Some research has tried to detect rogue APs by the delay from the client to the Internet server[2]. But current wireless LAN is so crowded especially in the 2.4GHz band, that delay fluctuates greatly.

In this paper, we propose a method to detect rogue wireless APs by comparing delay fluctuations of different backbone networks. To improve detection accuracy, we used the delay of a backbone network which is defined by subtracting the time taken for an ICMP packet to travel from client to first gateway from the time to the Internet server. Furthermore, we collected 100 samples of the backbone networks to normalize delay fluctuation and created a histogram based vector to represent a delay fluctuation in the backbone network. Finally, by comparing the current cosine distance of the delay histogram with the cosine distance of past legitimate connection delays histogram, our method judges whether the client is connected to a legitimate AP or rogue one. We evaluated whether backbone-delay-based rogue AP detection is effective within an experimental environment with six SSIDs. We treated one SSID as a legitimate AP and treated the other five SSIDs as rogue APs with different backbone networks. If the cosine distance between the current delay histogram and the delay histogram of the legitimate AP the previous day is longer than the threshold, it is considered the client has connected to the legitimate AP. The evaluation showed that by using a 0.8 threshold value, the proposal cannot detect rogue Aps on a backbone network closest to the legitimate AP since there are only slight differences between cosine distances. Thus, we have to consider an alternative method based on threshold-based definition. The evaluation result showed that detection accuracy could be improved by considering a weekday or and a weekend day as comparison targets. However, this method could only improve the detection slightly even if hours in a day and days of the week were taking into consideration.

The rest of the paper is organized as follows. Section II

overviews the related work. We discuss Evil Twin Attack and Rogue Access Points in Section III. The proposal for the detection method is detailed in Section IV. The environment setup is showed in Section V. We discuss the result in Section VI. Finally, we discuss the conclusion and future work in Section VII.

II. Related work

The cyber security of the Wi-Fi environment has been attracting researchers' interest, and there are several methods to detect ETA.

Wireless networks are usually identified by their SSID or MAC address. Thite et Al. proposed a method using the SSID, MAC address and RSSI to identify whether the AP is an authorized one [1]. However, the value of RSSI can be affected by environmental situation and all of the SSIDs, MAC addresses and RSSIs can be set the same as the real AP. Besides, there are other ways to make clients connect to the rogue AP apart from providing a stronger RSSI.

There are two kinds of ETA detections. One is the administrator-side detection and the other one is user-side detection.

Nakhila et Al. proposed a user-side approach relying on SSL/TCP [2]. They assumed that all of the APs in a hotspot use the same gateway to access the Internet. However, the attackers provide the Internet by using different gateway. Their method can detect whether the APs with the same SSID use different network gateways. However, they only considered a situation with one legitimate AP and one rogue AP and it will not be able to identify which one is the fake AP.

Bryah et Al. proposed a method using temporal traffic characteristics to detect the rogue AP from a central location independent of the wireless technology [3]. The authors focused on the differences in traffic characteristics when flows were from different sources. However, this method was limited: it would be costly if it was used in a large-scale network such as a campus network because, in this method, the switch where the rogue AP is immediately connected would be used to monitor the rogue AP.

Hao et Al. utilized RTT of DNS query between the client and the DNS server to distinguish a rogue AP [4,5]. However, there would be a lot of factors that might have influence on the result of the RTT .

Lu et Al. designed a system called ETD-SLFAT which can distinguish a rogue AP from lots of AP in a hotspot by comparing the arrival time of special length data frames [6]. One of the advantages of this detection is that the user does not need to connect to the potential rogue AP to know if there is a fake AP. However, they only considered situations wherein attackers use the same gateway as the legitimate AP.

Modi also considered a detection method on user-side [7]. They divided Evil Twin Attack into two situations – one is with same SSID and BSSID, and the other one with a different SSID and BSSID. When they are same, the external IP detects if it needs to warn the user. If the SSID and BSSID are different, one method to consider for detecting rogue Aps would be that

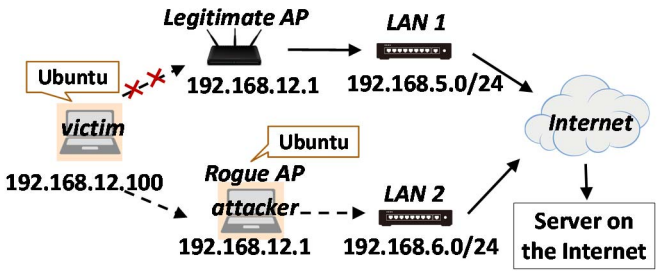

Figure 1. Evil twin attack with different backbone network

deauthentication frames broadcast which SSID or BSSID is usually used when the attacker wants to make user end disconnect from the legitimate AP.

Kumar et Al. made use of difference of authentication and association processes between the rogue and legitimate APs when the request frame was sent to these two APs from clients [8]. It also utilized both SSID and BSSID.

Hsu et Al. designed an ETA detector which can detect evil twins on the client side [9]. The ETA detector made an advantage of the packet-forwarding behavior in the TCP connection when rogue AP transports the packet from the victim to the Internet server. However, there are many ways to accomplish ETA, and attackers can use the mobile network to connect to the Internet server. And the ET detector is not able to detect the attack in this situation.

Wu et Al. proposed a method using received signal strength (RSS) to detect rogue APs [10]. Because the RSS vectors of beacon frames will change in different locations, the author utilized the changes in RSS information and took missing RSS values into consideration to reduce the risk of false alarms.

Zhou et Al. proposed a crowdsensing-based method of rogue AP detection [11]. The authors exploited the mobile crowd connection to a potential AP and made use of RSS as well.

As discussed above, information on the AP or traffic such as SSID, BSSID and delay could be useful. However, some authors only considered the situation that hotspots are only connected by a few APs and other only took relay-based rogue APs into consideration. Hence, in this paper, we have not only performed a series of experiments with the rogue AP which uses different backbone networks from legitimate networks, but we also contemplated the situation that attackers might use many types of backbone networks to create rogue APs. Besides this, most delay-based detections have failed to consider the effect of environment on the delay between client and AP. Therefore, we defined delay in a different way and proposed a novel delay-based detection method to distinguish rogue APs.

III. Evil twin attack and rogue access point

There are two types of implementation in ETA. One implementation is using a different backbone network implementation (different network) and the other is forwarding to the legitimate AP (forwarding). Figure 1 shows an implementation of ETA which is adopted in part of our research. In usual use, a client or victim machine is connected to the

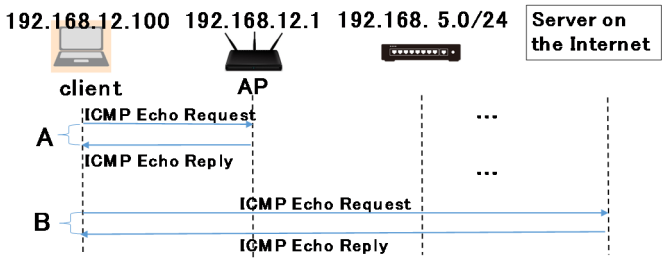

Figure 2. Backbone delay measurement method

legitimate AP. If a malicious person attempts an ETA, and prepares a rogue AP with an SSID and WPA password identical to the legitimate AP's, the victim client will receive a stronger signal from the rogue AP and connect to that. We prepared the environment, as shown in Figure 1 and confirmed that the rogue AP can capture traffic packets from the victim client with packet capture software (e.g. tcpdump). The environment shown in Figure 1 is implemented as different network types, but we can easily modify it to a forwarding-type implementation by replacing the "rogue AP to LAN2" path with a "rogue AP to legitimate AP" path. The forwarding type implementation explicitly extends backbone delay so that different network type implementations for the evaluation environment could be prepared.

An evil twin attack is not difficult; it can be realized by everyone. All you need is a laptop, an adapter used to create a fake AP and a dongle used for internet connection and some software to get information about the networks, such as the SSID or MAC address and so on.

IV. PROPOSAL OF DETECTION METHOD

As written in Section I, our proposal is based on the delay fluctuation of the backbone network. A rogue AP simulates the behavior of a legitimate AP, but it is difficult to set up the backbone network or the authentication server in backbone as a real one. In such a situation, it is considered that we can discriminate the rogue AP from the legitimate AP based on the delay of the backbone network. Therefore, we proposed a method that detects rogue APs by measuring delay fluctuation from the first network gateway to the Internet server with preliminary evaluation [12]. But in this study, we only showed cosine distance differences between delay histograms of different SSIDs and we did not define how to judge rogue APs.

In this paper, we propose a threshold based rogue AP detection based on cosine distance

Below, we present in detail a procedure to detect rogue APs.

Firstly, we measured the delay of the backbone network 100 times by subtracting the delay to the first network gateway from the delay to the Internet server. Detailed procedures are as followed.

1. Connect to the AP as client.
2. Send Ping (ICMP echo request) from the client to the first gateway (AP or router), as represented by A in Figure 2. Simultaneously, send ping from the client to the server on the Internet such as Google server, as represented by B in Figure 2.
3. Repeat step 2 100 times to collect the figures for delay A (from the client to the first gateway) and delay B (from the client to the Internet server).
4. Subtract delay A from delay B for each of the samples, respectively.

Table 1. Reconstruction of delay histogram

Delay [ms]	Bin width [ms]	Number of Data
delay< 0	-	1
0<= delay< 10	1	10
10<= delay< 30	2	10
30<= delay< 50	5	4
50<= delay< 100	10	5
delay>= 100	-	1

With the above procedures, we obtained 100 backbone delay samples and reconstructed these 100 backbone delay samples as a histogram-based vector with has 31 elements (Table 1).

Finally, we calculated the similarity of the vectorized backbone delay histogram with the current connected network and the previous connected network (Figure 3).There are several similarity calculation methods between vectors, and we chose cosine distance, which is one of the most well-known calculation methods. We firstly created a vectorized backbone delay histogram of the current connected network (V_{SSID1}) and the previous connected network (V_{SSID2}), respectively (Figure 3). Then, we calculated their cosine distances. If the cosine distance was larger than threshold, the backbone delay of the

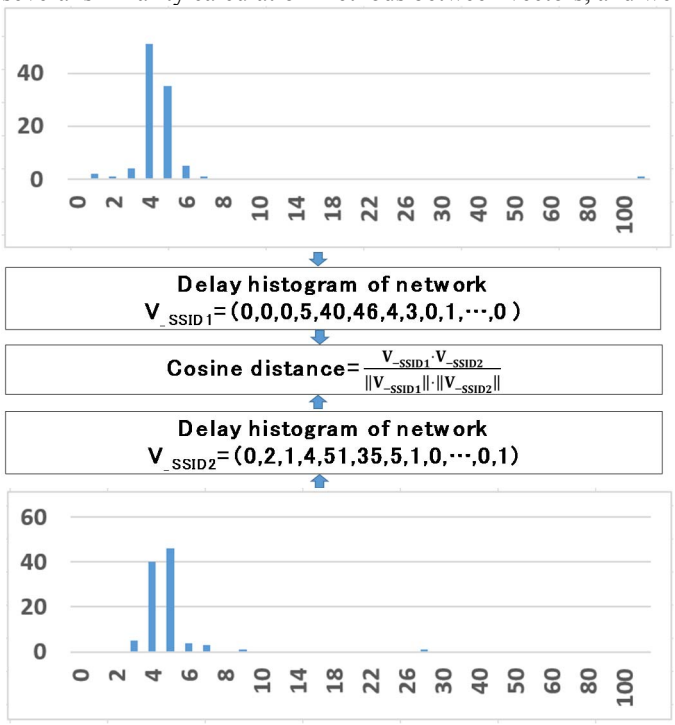

Figure 3. Calculation of cosine distance from vectored d elay histogram

current connected network is similar to the previous connected network, so we would define it as "connected to previous connected (not a rogue AP)". Otherwise, we treated it as "connected to different network", even if the SSID is an identical (possible rogue AP)". We defined the threshold value by evaluating the results of measuring the cosine distances between different networks shown in Section V.

In practical usage, delay histogram of legitimate AP was prepared beforehand and compared with delay histogram whenever there is new AP connected.

In this research, to improve the practical accuracy of the detection method, we adopted a rogue AP detection in a real-world hotspot where there are more than two APs. The multiple AP environments provided interference-related delay fluctuations between APs.

V. Experimental Setup

To define the threshold value for separating the same backbone network from different backbone networks, we evaluated the vectorized cosine distances over six different wireless networks every hour on different days (including different days of the week).

The SSID of six wireless networks were named legitimate, laboratory (Lab.), rogue, campus1, campus2, and mobile router. Figure 5 shows an outline of the backbone networks and AP differences among the six wireless networks. SSID mobile routers utilized cellular phone networks for backbone so that there were explicit differences in the backbone. SSID campus1 and campus2 are campus-wide Wi-Fi infrastructures. They shared the same AP for service, but the backbone networks are different because SSID campus2 did not function through the campus LAN. SSID legitimate, Lab., and rogue are internal laboratory wireless networks prepared for this evaluation. SSID legitimate and Lab. shared backbone networks so that only the AP maker was different. As result showed, at least one router or AP is different between these SSIDs. In addition, because it is a quite simple way to implement the rogue AP (wired LAN. SSID legitimate, Lab., and rogue are internal laboratory wireless networks prepared for this evaluation. SSID legitimate network and power supply are not essential), including a mobile router in the experiment is necessary.

In the environment of the laboratory where the four wireless networks were set up already, we set up one more wireless

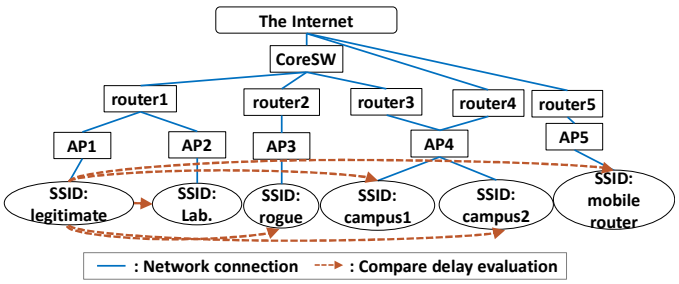

Figure 4. Backbone network difference between six SSIDs

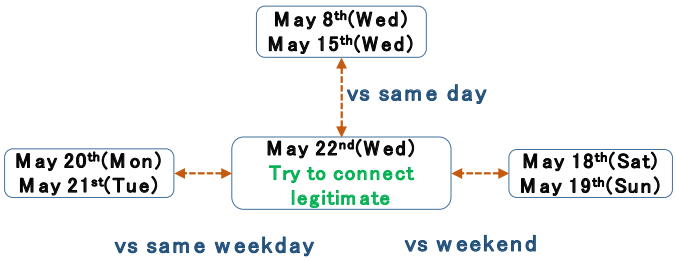

Figure 5. Comparison for day level difference

network whose SSID was legitimate and created a rogue AP on Ubuntu using a different gateway to connect to the Internet server and compared it with the legitimate AP. The rogue AP had the same SSID and IP address as the legitimate AP but had a different MAC address.

In this study, we selected the delay data of May 22, 2019 (a Wednesday) for the delay data of current connection and compared them with the data from earlier days. We chose May 8 and May 15 for comparison targets of earlier Wednesdays. We chose May 18 and May 19 for the closest earlier weekdays. We chose May 20 and May 21 for the closest earlier weekend days. We calculated cosine distances between the SSID legitimate and all SSIDs (red dotted arrow in Figure 4) from May 22 and the prior six days (Figure 5). Calculation pattern were shown in 36 patterns on the day level. Furthermore, we calculated cosine distance between every hour of the day, so in total 576 patterns were obtained.

VI. Result

A. Comparison Result among Every Hour During a Day

In this study, we considered May 22 (Wednesday) as the day on which the client tries to verify. We prepared delay data from the past day every hour so that there are 24 comparison targets even if we choose a comparison target day. Later, we compared results by changing the target hours. Firstly, we created a heat map of cosine distances between individual hours (Table 1). Table 1 showed the heat maps of the same SSID (campus1) on different days (May 18 vs May 22). The horizontal axis shows day and time of past day (e.g. "518 13" means May 18, 13:00) and the vertical direction shows day and time of verification. As a result, we can see an outlined time-varying characteristic of delay similarity. A slight, strong correlation between the same hours was identified (Table 2). Also, some irregular similarity drops due to irregular values were shown as comparison results from May 18, 14:00 or from May 22, 19:00. With these results, we prepared comparison candidates as shown below.

1. Comparing only the same hour (e.g. May 22, 13:00 vs May 15, 13:00)
2. Comparing the same hour with anteroposterior one hour (e.g. May 22,13:00 vs May 15,12:00/13:00/14:00) and taking the average
3. Comparing the same hour and anteroposterior two hours (e.g. May 22, 13:00 vs May 15, 11:00/12:00/13:00/14:00/15:00) and taking the average

Table 2 Cosine distance on same SSID (campus1) between each hour of May 22 and May 18

	518 00	518 01	518 02	518 03	518 04	518 05	518 06	518 07	518 08	518 09	518 10	518 11	518 12	518 13	518 14	518 15	518 16	518 17	518 18	518 19	518 20	518 21	518 22	518 23
522 00	0.8236	0.7308	0.8244	0.8387	0.8040	0.9170	0.7091	0.8481	0.8227	0.8089	0.7801	0.8252	0.9033	0.9099	0.3505	0.8086	0.8487	0.8648	0.9146	0.8370	0.7743	0.7355	0.7308	0.8556
522 01	0.9651	0.9407	0.9244	0.9384	0.9276	0.8114	0.8930	0.9111	0.9140	0.9203	0.8849	0.9068	0.7755	0.7287	0.4334	0.8250	0.4660	0.8611	0.7507	0.9178	0.9125	0.9665	0.9606	0.9221
522 02	0.8461	0.8052	0.8920	0.8583	0.8901	0.8908	0.7304	0.8789	0.8461	0.8163	0.8323	0.8440	0.8788	0.9102	0.2817	0.8834	0.7965	0.9529	0.8903	0.9085	0.8354	0.8247	0.8243	0.9315
522 03	0.9275	0.8840	0.9239	0.9568	0.9157	0.8998	0.9329	0.9300	0.9368	0.9556	0.8725	0.9467	0.8859	0.8059	0.3586	0.9307	0.6312	0.8978	0.8556	0.8385	0.9596	0.9078	0.9215	0.9216
522 04	0.8246	0.7827	0.8745	0.8157	0.8299	0.8732	0.6635	0.8759	0.8361	0.7702	0.8058	0.7792	0.7933	0.9261	0.3208	0.7605	0.7733	0.9093	0.8797	0.9290	0.7547	0.8020	0.7890	0.8623
522 05	0.8959	0.8707	0.9150	0.9720	0.9057	0.8945	0.9379	0.8919	0.8932	0.9304	0.9107	0.9534	0.9002	0.7890	0.3113	0.9547	0.6144	0.8962	0.8218	0.8454	0.9420	0.8739	0.8930	0.9172
522 06	0.8162	0.7992	0.8833	0.9082	0.9199	0.8881	0.8672	0.8505	0.8694	0.9205	0.8343	0.9206	0.8783	0.7833	0.4146	0.9301	0.6841	0.9280	0.8654	0.8125	0.8943	0.8307	0.8588	0.9246
522 07	0.7901	0.7074	0.8754	0.8577	0.7653	0.9452	0.7444	0.8808	0.8622	0.8372	0.7751	0.7990	0.9366	0.9625	0.2288	0.8539	0.8905	0.9010	0.9361	0.7841	0.8107	0.7358	0.7405	0.8180
522 08	0.8371	0.8098	0.9219	0.9467	0.8873	0.9416	0.9092	0.9008	0.9123	0.9533	0.8652	0.9330	0.8779	0.8230	0.4345	0.9007	0.6773	0.9208	0.8897	0.8342	0.9013	0.8451	0.8566	0.8744
522 09	0.9189	0.8814	0.9608	0.9549	0.8422	0.9362	0.8928	0.9494	0.9624	0.9452	0.9018	0.8863	0.8750	0.8912	0.3592	0.8507	0.6725	0.8923	0.8791	0.8666	0.8858	0.8756	0.8779	0.8713
522 10	0.9108	0.9260	0.8948	0.9322	0.9181	0.8063	0.9632	0.9010	0.9315	0.9390	0.9037	0.9388	0.7724	0.6755	0.3893	0.9015	0.4300	0.8288	0.7317	0.8032	0.9563	0.9234	0.9288	0.8815
522 11	0.9228	0.9303	0.9483	0.9438	0.8946	0.8318	0.9628	0.9321	0.9526	0.9525	0.9002	0.9219	0.7879	0.7315	0.3319	0.8983	0.4612	0.8766	0.7490	0.8222	0.9580	0.9385	0.9523	0.8772
522 12	0.8829	0.8297	0.9162	0.9051	0.8442	0.9595	0.8162	0.9177	0.9248	0.9131	0.8576	0.8622	0.8813	0.9194	0.4495	0.8233	0.7572	0.8928	0.9400	0.8819	0.8302	0.8196	0.8197	0.8723
522 13	0.9032	0.8536	0.9183	0.8733	0.8493	0.9089	0.7651	0.9354	0.9044	0.8539	0.8496	0.8481	0.8537	0.9330	0.3013	0.8327	0.7572	0.9130	0.9003	0.9071	0.8407	0.8468	0.8437	0.8895
522 14	0.9372	0.9048	0.9866	0.9538	0.8746	0.8812	0.9247	0.9624	0.9603	0.9418	0.8776	0.9008	0.8118	0.8207	0.2881	0.8746	0.5567	0.9228	0.8042	0.8612	0.9419	0.9503	0.9520	0.8759
522 15	0.8896	0.8186	0.9371	0.9283	0.8461	0.9624	0.8669	0.9335	0.9296	0.9249	0.8533	0.8835	0.9181	0.9053	0.3425	0.8838	0.7576	0.9183	0.9147	0.8457	0.8886	0.8430	0.8451	0.8734
522 16	0.9469	0.9270	0.9207	0.9711	0.9295	0.8914	0.9533	0.9263	0.9452	0.9648	0.9191	0.9587	0.8501	0.7654	0.4640	0.8986	0.5552	0.8664	0.8269	0.8806	0.9387	0.9138	0.9201	0.9237
522 17	0.8337	0.7714	0.9014	0.8965	0.8811	0.8874	0.7916	0.8763	0.8445	0.8474	0.7975	0.8653	0.9212	0.8680	0.2125	0.9359	0.7851	0.9581	0.8651	0.8467	0.8940	0.8357	0.8464	0.9188
522 18	0.8537	0.8148	0.8896	0.9282	0.8645	0.9675	0.8346	0.8726	0.8911	0.9171	0.8955	0.8891	0.8945	0.8901	0.5206	0.8293	0.7382	0.8727	0.9481	0.8984	0.8140	0.8038	0.8019	0.8854
522 19	0.5723	0.5039	0.7125	0.6888	0.6216	0.8850	0.5261	0.6870	0.6830	0.6873	0.6404	0.6482	0.8354	0.9153	0.3189	0.6779	0.9479	0.7991	0.9254	0.6848	0.5647	0.5016	0.518 1	0.6952
522 20	0.7212	0.6780	0.7643	0.8167	0.8137	0.9326	0.7016	0.7727	0.7769	0.8073	0.7830	0.8240	0.9064	0.8799	0.4850	0.8257	0.8666	0.8447	0.9621	0.8225	0.7272	0.6522	0.6678	0.8512
522 21	0.8904	0.8746	0.8661	0.9261	0.9464	0.8488	0.8985	0.8578	0.8502	0.8894	0.8852	0.9633	0.8467	0.7355	0.3641	0.9454	0.5790	0.8878	0.8012	0.8653	0.9234	0.8746	0.8960	0.9500
522 22	0.7177	0.6497	0.8437	0.8142	0.7365	0.9433	0.6969	0.8297	0.8257	0.8129	0.7337	0.7691	0.8898	0.9446	0.3047	0.8118	0.9007	0.8941	0.9507	0.7546	0.7414	0.6703	0.6793	0.7904
522 23	0.9211	0.9110	0.8825	0.9296	0.9330	0.8210	0.9100	0.8549	0.8704	0.9141	0.9043	0.9414	0.8732	0.7248	0.3550	0.9524	0.5324	0.8618	0.7659	0.8399	0.9325	0.8862	0.9048	0.9559

0.95<=cos.dist.<=1.00 0.90<=cos.dist.<0.95 0.85<=cos.dist.<0.90 0.80<=cos.dist.<0.85
0.75<=cos.dist.<0.80 0.70<=cos.dist.<0.75 0.50<=cos.dist.<0.70 0.00<=cos.dist.<0.50

4. Comparing whole hours of the day and taking the average

We prepared the averaged cosine distances to normalize irregular values based on same-hour comparisons. However, the result did not differ so largely (less than 1 percent) among methods 1 To 4. The results were 0.8353, 0.8371, 0.8332, and 0.8353, respectively.

This is because method 1 affected the irregular value largely due to limited comparison results, but method 4 had less effect on irregular values, so there was only a small difference. This trend was also seen in other comparison results, so we finally chose method 4 for the latter analysis in order to screen out the irregular values.

While discussing this with the heat map, we sometimes saw results with quite high similarity for the whole day and sometimes significantly less similar results. The prior result is shown from comparing the SSID legitimate of May 8 and the SSID legitimate of May 15. An averaged similarity was 0.9556. The latter result was shown of comparing SSID mobile routes and arbitrary SSIDs on an arbitrary day. The average similarity is 0.

B. Cosine Distances between the Same SSID and a Different SSID

To define a threshold to judge whether the connected SSID is identical to a previous connected one, we compared cosine distance distributions between the same SSID and different SSIDs on different days. Figure 6 shows the histogram of averaged cosine distances between every hour per day. The horizontal axis showed the count per bin. It explicitly showed that the similarity of the same SSID is greater than of the different SSIDs in major cases. There was low similarity, even in the same SSID, of around 0.2. This is a comparison result between different days on the SSID mobile router. This result may come from the delay instability of the cellular phone

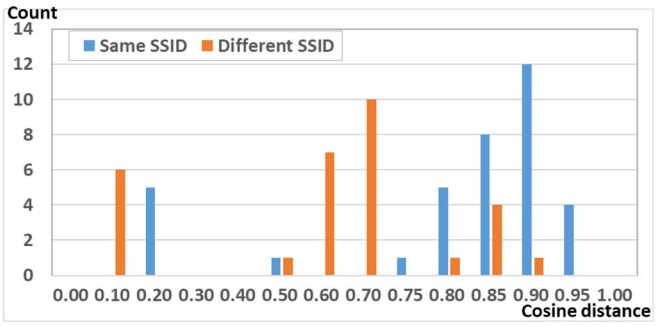

Figure 6 Histogram of cosine distances between 2 types of SSID

network. On the other hand, six different SSID results gave more than 0.8 similarity. Those results were similar between SSID legitimate and SSID Lab., the two with the closest backbone networks (Figure 6). Finally, according to Figure 6, we defined the threshold at 0.74, even if it did not recognize the difference between SSID legitimate and SSID Lab..

For more discussion, we prepared Table 3 that showed the average of the cosine distances between SSID legitimate on May 22 and the other SSIDs on different types of days. Firstly, we separated comparison target to the same days of the week (1 week before and 2 weeks before), the closest weekdays, and the closest weekend days. As shown from the legitimate column of Table 3, similarity between same days of the week is worse than that of the closest weekdays. Similarly, there is a low similarity on weekend, including similarity to different SSIDs on weekend. Thus, we defined that the comparison target delay data should be separated into weekdays and weekend days.

As mentioned before, it is hard to separate SSID legitimate and SSID Lab. (Figure 6). For further discussion, we prepared Table 3, which showed details of the similarities between SSID legitimate and SSID Lab. on individual days. Similarities

Table 3. Average of cosine distance between legitimate a nd other SSIDs

legitimate	mobile router	campus1	Lab.	legitimate	rogue	campus2
May 8, 15(Wed)	0.0000	0.3125	0.9375	0.9375	0.0000	0.1250
May 18, 19(weekend)	0.0000	0.1458	1.0000	0.9167	0.0000	0.1875
May 20, 21(weekday)	0.0000	0.4375	0.9583	0.9583	0.3333	0.4375

Table 4. Cosine distance between the same backbone ne tworks

		laboratory vs legitimate	legitimate vs legitimate
Wed. vs Wed.	May 15 vs 22	0.7955	0.8821
	May 8 vs 22	0.8440	0.8535
Wed. vs weekday	May 20 vs 22	0.8556	0.8508
	May 21 vs 22	0.7918	0.8559
Wed. vs weekend	May 18 vs 22	0.8345	0.8803
	May 19 vs 22	0.8304	0.7810

between SSID legitimate and itself are greater than those between SSID legitimate and SSID Lab. excluding May 19 vs May 22 and May 20 vs May 22. May 19 vs May 22 can be omitted by separating weekdays and weekend days, but it cannot separate out May 20 vs May 22 with the current method (Table 3). However, it seems that there are some similarity advantages with SSID legitimate so that there is possibility of separate them by improving the similarity calculation and threshold value definition method.

C. Detection Accuracy

Based on the threshold value defined in Section V-B., we evaluated rogue AP detection accuracy. In this evaluation, firstly, we assumed that the client tries to connect SSID legitimate on May 22, and there was a possible rogue AP that showed a fake SSID legitimate which had same backbone delay as the other five SSIDs. Based on a 0.74 threshold value and adopting method 4 (compared with whole hours of the day and taking the average), we detected rogue APs with 67.50 percent accuracy with a false positive rate of under 4.86 percent (rated as a rogue AP even if connected to a legitimate AP). The threshold value obtained from the explicit separation point in Figure 6 gave less false positives, with less accuracy. We also evaluated the proposed method with a 0.85 threshold value which was observed from the heat map (Table 1). The result showed that rogue Aps were detected with 86.67 percent accuracy with a 6.25 percent false positive rate.

VII. Conclusion and Future Work

Wireless networks play as an important role in everyday life. We rely on Wi-Fi in so many ways because of its convenience such as business and personal life. However, we must not ignore the risks of network security, that if attacked by a hacker on a wireless network, one might not be able to bear the loss.

In this study, we focused on the Evil Twin Attack which is a common risk on Wi-Fi. We proposed a method to detect the rogue AP by using delay fluctuations on the backbone network. This method could recognize an Evil Twin Attack when a rogue AP uses different gateways.

Our results from the experiment showed that our method can be useful for detecting rogue APs that use different gateways with totally different backbones. However, our method does not function well when encountering the same backbone, like with SSID Lab. and SSID legitimate. Our method also identified a slight difference between the same backbones.

As for future work, we should look for a method to identify which network the users are currently connected to, such as SSID Lab. and SSID legitimate, with long term data. Eventually, we plan to create a tool that can evaluate and compare delays and warn the end users if the currently connected AP is possibly a rogue AP.

Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Number 19H04108 and 19K11961.

References

[1] S. Thite, S. Vanjale, and P. B. Mane, “A Novel Approach for Fake Access Point Detection and Prevention in Wireless Network,” International Journal of Computer Science Engineering and Information Technology Reaserach, pp. 35-42. Feb. 2014.

[2] O. Nakhila, E. Dondykc, M. F. Amjadd, and Cliff Zoue, “User-Side Wi-Fi Evil Twin Attack Detection Using SSLTCPProtocols,” In Proc. of Consumer Communications & Networking Conference 2015, pp. 239-244, Jan. 2015.

[3] R. Beyab, “Rogue Access Point Detection using Temporl traffic characteristics,” In Proc. of Global Telecommunications Conference 2004, vol. 4, pp. 2271-2275. Dec. 2004.

[4] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engle, “Undesired Relatives: Protection Mechanisms Against The Evil Twin Attack I IEEE 802.11,” In Proc. of the 10th ACM symposium on QoS and Security for Wireless and Mobile Networks, pp. 87-94, Sep. 2014.

[5] H. Han, B. Sheng, Chiu C. Tan, Q. Li, and S. Lu, “A Timing-Based Scheme for Rogue AP Detection,” IEEE Transactionson Parallel and Distributed Systems, Vol. 22, Issue 11. pp. 1912-1925. Nov. 2011.

[6] Q. Lu, H. Qu, Y. Ouyang, and J. Zhang, “SLFAT:Client-Side Evil Twin Detection Approach Based on Arrival Time of Special Length Frames,” Security and Communication Networks, Vol. 2019, pp. 1-10. Jun. 2019.

[7] V.Modi, “Detection of Rogue Access Point to Prevent Evil Twin Attack in Wireless Networks,” International Journal of Engimeering Research and Technology, Vol.6 Issue 4, pp. 23-26, Mar. 2017.

[8] A. Kumar, B. Raj, and P. Paul, “Detection and Prevention against Evil Twin Attack in WLAN, ” International Journal of Computer Engineering and Application, Special Edition, ISSN 2321-3469, Aug. 2016.

[9] F. –H. Hsu, C.-S. Wang, Y.-L. Hsu, Y.-P. Cheng, Y.-H. Hsneh “A Client-Side Detection Mechanism for Evil Twins,” Computers and Electrical Engineering, Vol. 59, pp. 76-85, Nov. 2015.

[10] W. Wu, X. Gu, K. Dong, X. Shi, and M. Yang, “PRAPD: A Novel Received Signal Strength-Based Approach for Practical Rogue Access Point Detection,” International Journal of Distributed Sensor Networks, Vol. 14, No.8, Aug. 2018.

[11] T. Zhou, Z. Cai, B. Xiao, Y. Chen, and M. Xu, “Detecting rogue AP with the crowd wisdom,” In Proc. of the 37th International Conference on Distributed Computing Systems, pp. 2327-2332, Jun. 2017.

[12] Z. Zhang, H. Hasegawa, Y. Yamaguchi and H. Shimada, “Rogue Wireless AP Detection using Delay Fluctuation in Backbone Network,” In Proc. of the 43rd Annual International Computers, Software and Applications Conference, pp. 936-937, Jul. 2019.