Letting the Puss in Boots Sweat: Detecting Fake Access Points using Dependency of Clock Skews on Temperature

Fabian Lanze
University of Luxembourg
fabian.lanze@uni.lu

Andriy Panchenko
University of Luxembourg
andriy.panchenko@uni.lu

Benjamin Braatz
University of Luxembourg
benjamin.braatz@uni.lu

Thomas Engel
University of Luxembourg
thomas.engel@uni.lu

ABSTRACT

The only available IEEE 802.11 network identifiers (i. e., the network name and the MAC address) can be easily spoofed. Consequently, an attacker is able to fake a real hotspot and attract its traffic. By this means, the attacker can intercept, collect, or change users' traffic (often even if it is encrypted).

In this paper, we describe an efficient method for detecting the replacement of access points (APs) by passive remote physical device fingerprinting. The main feature of our fingerprinting approach is the clock skew—an unavoidable phenomenon that causes clocks to run at minuscule yet remotely observable different speeds—which is extracted from information contained in beacon frames. We are the first to achieve a high discriminability of devices by completely eliminating the fingerprinters' influence and considering the clock skew's dependency on temperature.

Finally, we develop a method for reliable detection of the presence of AP impostors that works without explicit temperature information. Compared to the best state-of-the-art approach, our method improves detection accuracy from about 30 % to 90 % without generating any traffic and requires less than one minute to collect a sufficient number of observations. Our approach yields a strong feature for passive remote physical device fingerprinting in wireless networks.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*; C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations—*Public networks*

Keywords

Evil Twin Attack; Security; Wireless Access Point; Fake Access Point; 802.11; Device Fingerprinting; Clock Skew

ASIA CCS'14, June 4–6, 2014, Kyoto, Japan.
Copyright 2014 ACM 978-1-4503-2800-5/14/06 ...$15.00.
http://dx.doi.org/10.1145/2590296.2590333.

1. INTRODUCTION

In recent years, Internet usage shifted from fixed locations to mobile environments. Nowadays, people are used to being online all the time, independent of their location, using laptops, smartphones, or tablets. Wi-Fi access points (APs) offer fast and cost-effective Internet connectivity. They are available almost everywhere, in offices, on university campuses, and in public places such as cafés, shopping malls, hotels, or airports. Although mobile cellular networks (e. g., 3G) have gained an increasing influence, the importance of Wi-Fi networks remains crucial. Generally, they provide faster connectivity, offer service whenever mobile networks are unavailable, overloaded, or overpriced (e. g., in roaming). They are indispensable for devices that do not have hardware to access mobile cellular networks, e. g., laptops or many tablets.

The only identifiers provided by the IEEE 801.11 standard for a user to verify the authenticity of an AP are its SSID (i. e., the network name) and its BSSID (i. e., the MAC address). Since these can easily be spoofed, an attacker is able to fake an AP without the user being able to notice (in the literature/media, this attack is also know as *evil twin* [23, 9] and has gained recent attention[1]).

Once a user is connected to a faked AP, the attacker can mount various attacks, including interception, collection, or manipulation of transmitted data. This even remains possible if the user explicitly enables encryption, e. g., by using SSL. Since the attacker already established his AP as intermediary, he can easily act as man-in-the-middle. Nowadays, this does not require special skills as deployable tools such as SSLstrip[2] (which transparently removes SSL encryption) and BurpProxy[3] (which can create faked certificates on-the-fly) are freely available and easy to use. Since most users incautiously accept unsigned or wrongly-signed SSL certificates [26, 6], malicious access points are able to conduct man-in-the-middle attacks on encrypted traffic (i. e., can read and modify the data) and to hijack sessions.

The danger of the described attacks arises from their simplicity: All common mobile operating systems including Android and iOS are capable of creating a wireless AP. Hence, this process can be be performed directly from smartphones,

[1] `http://www.abc15.com/dpp/money/consumer/alerts/alert-thieves-create-fake-hotel-wi-fi-hot-spots-to-steal-your-information`

[2] `http://www.thoughtcrime.org/software/sslstrip/`

[3] `http://portswigger.net/burp/proxy.html`

without attracting the attention of anybody in the vicinity. Additionally, fully-automated tools are available that are capable of spoofing APs, e. g., *rfakeap*[4], *airsnarf*[5], *jasager*[6].

Hence, we focus on attackers who mount short-duration evil twin attacks by using off-the-shelf hardware. Scenarios where an attacker needs to extensively modify hardware components are therefore out of the scope of this work.

The methods described in the 802.11 Robust Secure Network Association (RSNA) specification propose to solve this problem with cryptographic primitives using an additional authentication server [14]. This requires careful setup and maintenance. Operators of open hotspots in particular have no incentive to deploy such a mechanism and, hence, the solution is rarely used. The usage of virtual private networks (VPNs) also does not provide a satisfactory solution to the problem as, besides similar certificate-based attacks like those on SSL, it is possible to kill the VPN session (e. g., by dropping management packets) so that the connection falls back to plain mode, typically without explicit notification.

Many public hotspots deploy a web-based authentication scheme which is additionally used for accounting of clients. Nevertheless, such a mechanism does not provide any security for the user at all. The attacker could simply clone the login page and accept any credentials. Furthermore the attacker learns the legitimate credentials provided by users for further misuse. If pre-shared key (PSK) based encryption (such as WPA) is applied and the attacker knows the PSK, he can still mount the attack. Note that for public hotspots the PSK must be distributed to clients by some means or another (e. g., on a receipt).

Therefore, there is a strong need to equip users with additional tools and methods for verification of the APs they connect to, in order to make sure that these are authentic and not traps operated by an attacker. In our targeted solution, a user has access to a trusted third party that verifies the fingerprint of an authentic hotspot. Consequently, there is a vital necessity for tools that enable remote device identification and can be applied to currently deployed networks without modifying standardized protocols and without requiring the AP operator to cooperate or modify the deployment.

In this paper, we focus on remote device fingerprinting based on *clock skews*. This is an unavoidable physical phenomenon that causes crystal oscillator based clocks to run with minuscule yet measurable deviations in speed. The frequencies of crystal oscillators are determined by their manufacturing properties (e. g., the cut angle) and the crystal type. Inevitable inaccuracies during the production process lead to slight variations of the frequency, even for crystals produced within the same series [11].

Clock skew is based purely on physical properties, which makes it an attractive feature for physical device fingerprinting. It is typically measured in *parts per million (ppm)*.

For measuring the clock skew of a remote clock, it is essential to have access to precise timestamps generated by that clock. In the Wi-Fi scenario we benefit from the fact that APs emit management frames (called *beacons*) at a high frequency. These frames contain a high-precision timestamp with microsecond resolution, which is used for the Timing Synchronization Function (TSF) to synchronize sending and receiving slots on the shared wireless medium in the IEEE 802.11 communication process. All stations in a basic service set (BSS), i. e., all clients associated with an AP, synchronize their local TSF timer to the time broadcasted in beacon frames. Beacon frames are usually sent every 100 ms. By specification, they are not subject to any delay before sending. Therefore, they serve as a reliable basis for clock skew estimation.

Contribution:

We present a novel approach to reliably detect a spoofed access point utilizing the fact that, in general, other APs are simultaneously reachable in the vicinity. In detail, our contributions are as follows:

- We are the first to intensively study the influence of changing room temperatures on APs' clock skews in the context of fingerprinting. As we will show, APs exhibit significantly different temperature dependency characteristics. By learning these, the discriminability of clock-skew-based fingerprints (in terms of recognition interval sizes) can be improved by more than 75 %.
- We completely eliminate the influence of the fingerprinter (both skew and its dependency on temperature) on the fingerprint. This allows comparison of fingerprints produced by arbitrary fingerprinters. We substantiate this claim mathematically.
- We develop a method for fake AP detection which exploits temperature dependency without the need for explicit temperature knowledge and which provides similar accuracy. Here, a manageably small number of around 50 training observations is sufficient.
- Finally, we are the first to propose a practical solution to detect faked APs with a high probability.

2. RELATED WORK

Due to the severity of the underlying problem, the field of *remote physical device fingerprinting* has attracted considerable research interest in recent years and several methods have been proposed. We distinguish between *active* and *passive* techniques.

Active techniques explicitly interact with the device that is fingerprinted, e. g., by sending regular or manipulated packets and evaluating the response. Inherently, such methods can be detected by the fingerprintee or even cause interference with the regular communication. Sieka [25] evaluates timing patterns in the authentication procedure of an AP using two different measuring devices for fingerprinting. Bratus et al. [3] propose a technique called active behavioral fingerprinting, which is based on malformed stimuli response, i. e., how devices react to non-standard and malformed 802.11 frames.

Passive methods, on the other hand, do not require any cooperation with other nodes as they merely observe passing traffic without interaction or modification. Hence, these methods are by design undetectable and do not interfere with the regular communication. Therefore, we direct our focus to passive techniques.

[4] `http://rfakeap.tuxfamily.org/`

[5] `http://airsnarf.shmoo.com/`

[6] `http://www.digininja.org/jasager/`

The accuracy of remote wireless device fingerprinting can be optimized to more than 99 % if dedicated hardware is applied. Using, e. g., *radio frequency fingerprinting (RFF)* [24, 10, 4] such techniques investigate physical properties of the radio signal with specialized measurement devices. However, these methods are not applicable in our context as we target methods that work on regular mobile clients, which are not equipped with such hardware.

Another class of passive approaches does not aim to identify unique devices (APs) but rather unique device types [8] or device driver types [7]. Those methods do not satisfy our requirement of differentiating between APs with the same hardware and firmware.

Bahl et al. [2] identify the presence of two APs with the same BSSID. The method uses sequence numbers in 802.11 frame headers but is not able to recognize which of the APs is faked and is only applicable if both APs are active simultaneously in a nearby location. Gonzales et al. [9] present two methods to protect against the evil twin attack. The first method (EAP-SWAT) requires modification of the deployed authentication protocols which, in general, is undesirable as all deployed APs would have to be adapted. The second method (context-leashing) aims to detect whether the set of simultaneously visible APs has significantly changed. It achieves a comparably high detection accuracy. Note that this method can only detect a specific variant of the attack, where the evil twin AP is set up at a different location than the legitimate AP. Neumann et al. [21] analyze fingerprints based on network parameters such as transmission time and frame inter-arrival time. The best recognition rates for wireless devices are 40%–60% in a laboratory setting and 20%–32% in real-world traces.

A more promising field of research for wireless fingerprinting utilizes the phenomenon called *clock skew.* Originating from the work of Moon et al. [20], Kohno et al. [17] introduce the concept of clock skew based remote device fingerprinting using the TCP Timestamps option in TCP headers [13] or ICMP packets, both having timing resolution in milliseconds. They estimate the clock skew with linear programming (LP). Clock skews are shown to be distinguishable among different physical machines yet stable over time. The approach requires support of the timestamp option and observing a TCP connection or ICMP packets over a longer period of time, while wireless APs cannot be directly accessed via TCP/ICMP in general.

In [27] clock-skew-based fingerprinting is applied to wireless sensor nodes (WSN). The results are not comparable to our work as the authors assume that the fingerprinter is always the same (sink node) and can be kept in a constant temperature environment. This does not hold true in our scenario. Yang et al. [30] analyze the impact of temperature on clock skew estimation to improve clock synchronization for WSNs and the authors observe that two clocks exhibit a stable relationship of clock skew w.r.t. temperature which is not necessarily linear. This complies with our findings regarding TSF clocks (see Section 6). However, unlike in our scenario, for WSNs there is no necessity to model this temperature dependency fingerprinter-independent.

Jana et al. [15] transfer the idea of Kohno et al. to the 802.11 scenario, estimating APs' clock skews from TSF timestamps in beacon frames. Instead of the LP method the authors use a least squares fit estimation (LSF), which is more efficient but also more sensitive to outliers, since fewer outliers compared to timestamps in TCP packets are to be expected. Moreover, the use of TSF data requires significantly smaller sample sizes due to higher clock resolution. To measure the receiving time of a beacon frame, a modified driver is used. The authors argue that it is not possible to fake the clock skew using software alone because of unpredictable sending delays due to Medium Access Control. In their work, fingerprints are not comparable between different fingerprinter machines (FPs) due to the influence of the fingerprinting device's own clock skew. Arackaparambil et al. [1] improve the accuracy by utilizing the FP's TSF timer as more precise clock source. However, their technique does not remove the skew of the fingerprinter card from the clock skew estimation and is only evaluated with two Wi-Fi chipsets of the same type. In our previous work [18], we presented a lightweight method for clock skew estimation based on TSF timestamps in beacon frames using an online version of the LSF estimation with high precision. Our tools do not rely on modified drivers or system components and therefore enable clock skew estimation for APs from arbitrary mobile clients. An NTP based method was presented to eliminate the influence of the fingerprinter's skew. However, clock skew variations of approx. ±1ppm at different measuring times remained. Our survey of 388 APs shows roughly a normal distribution of clock skews within a range of -30 to 30 ppm. We concluded that the information content of clock skews alone is too limited and unique device fingerprinting using clock skews is not practically feasible for wireless access points due to its limited distinctiveness. Note, however that our evaluation did not consider the actual influence of environmental conditions such as temperature.

In general, no approach for wireless device fingerprinting has been able to identify 802.11 APs with acceptable detection ratios, while being performed from arbitrary mobile clients. We address this challenge and show that by including temperature information and applying state-of-the-art machine learning techniques, we are able to reliably detect the presence of faked APs. Moreover, we completely exclude the influence of the fingerprinter on the fingerprint. Our method successfully detects the replacement of an AP on a representative data set with an accuracy of 90%, thus, 60% better that the best approach known so far. This is done without actively generating any traffic and requires less than one minute to collect a sufficient number of observations.

3. CLOCK SKEW BACKGROUND

In this section, we describe physical and mathematical background of clock skews in time measuring hardware.

3.1 Quartz Oscillator Fundamentals

Common computer clocks (and thus the ones that operate the TSF timer in Wi-Fi chipsets) are based on crystal oscillators. The major component of a crystal oscillator is an anisotropic crystal formed from SiO_2 (quartz). Due to its piezoelectric properties a mechanical strain is produced in the crystal when exposed to an electric field. The resulting vibration enables the crystal to be used as resonator in an oscillator unit that can generate clock signals. Although other types of resonators exist, quartz has proven to be superior regarding the properties frequency stability, intrinsic loss, simplicity of production and cost. Hence, since many years, quartz oscillators have been the preferred medium satisfying the needs for precise frequency generation [16].

Crystal oscillator stability and accuracy is affected by physical and electrical factors, with temperature being one of the most significant. The dependency on temperature is primarily determined by the angle at which the crystal is cut with respect to the crystallographic axes during the manufacturing process [29]. Most common is the AT-cut. This crystal type exhibits a cubic dependency on temperature [31]. The turning point of this cubic function and the local extrema can be controlled by small variations of the cut angle. The inflection point is typically located at a temperature of 25 °C (77 °F) leading to highest stability at typical operating temperatures. More than 90% of all deployed crystal oscillators are based on an AT-cut crystal.

The described properties of crystal oscillators enable the vendors of Wi-Fi chipsets to select units that meet given specifications. For 802.11 devices, a frequency tolerance of ±25ppm is specified [12].

3.2 Mathematics of Clock Skews

We model (discretized) *true time* by the set $\mathbb{Z}$ of integers with an arbitrarily chosen zero point and measured in arbitrarily chosen units. A *clock* counts time steps from a (possibly different) zero point, i. e., it is a function $C\colon \mathbb{Z} \to \mathbb{Z}$, giving the mapping from true time t to the time $C(t)$ measured by the clock. Without loss of generality, we assume that all clocks have the same time step resolution as the unit chosen for true time—microseconds in our case[7].

For an ideal clock, the difference $C(t_2)-C(t_1)$ between two time measurements at points t_1 and t_2 in true time would always be $t_2 - t_1$. However, due to physical properties (see Section 3.1), clocks based on crystal oscillators exhibit a certain *offset*

$$\mathrm{offset}_C(t_1, t_2) = [C(t_2) - C(t_1)] - [t_2 - t_1] \qquad (1)$$

when measuring such time intervals (positive if the clock is too fast and negative if the clock is too slow). The *skew* s_C of a clock C between points t_1 and t_2 in true time is the slope of the offset between these points:

$$\mathrm{s}_C(t_1, t_2) = \frac{\mathrm{offset}_C(t_1, t_2)}{t_2 - t_1} = \frac{C(t_2) - C(t_1)}{t_2 - t_1} - 1 \qquad (2)$$

Typically, it is not possible to measure the clock skew directly, since the fingerprinter's clock has its own non-negligible skew. Hence, there are two clocks: the clock C of the access point and the clock D of the fingerprinter. We observe both at points t_1 and t_2 in true time and compute a *subjective offset*

$$\mathrm{offset}_{D,C}(t_1, t_2) = [C(t_2) - C(t_1)] - [D(t_2) - D(t_1)] \qquad (3)$$

and a *subjective skew*:

$$\mathrm{s}_{D,C}(t_1, t_2) = \frac{\mathrm{offset}_{D,C}(t_1, t_2)}{D(t_2) - D(t_1)} = \frac{C(t_2) - C(t_1)}{D(t_2) - D(t_1)} - 1 \qquad (4)$$

Figure 1 shows the offsets of two different access point clocks measured from two different fingerprinters at two different measuring times. The fingerprinters' skews lead to significantly different subjective skews for the same access point clocks. Moreover, we observe a small difference for the different measuring times.

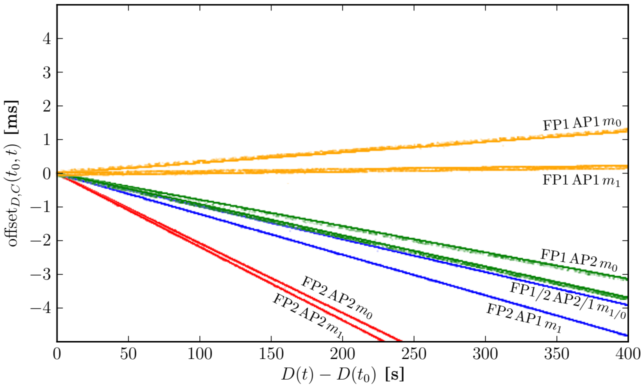

Figure 1: Subjective offsets of the clocks of two different access points (C = AP1 and C = AP2) measured from two different fingerprinters (D = FP1 and D = FP2) at two different measuring times ($t_0 = m_0$ and $t_0 = m_1$)

We can already see that (at least for short periods of time—in the range of several minutes) the offsets increase or decrease linearly. To overcome noise produced, e. g., by the communication hardware and the device drivers, we not only take the measurements at two border points (as in the mathematical formulae), but use all available measurements within a certain interval (several minutes for high precision). The skew is then approximated by the slope of a least squares fit (LSF) linear regression over these measurements. LSF has shown to be the superior method for the slope estimation in the described setting [18, 15].

The goal of the following sections is to eliminate the influence of the fingerprinters and model the influence of temperature on the access points in order to isolate the differences between access points and use them as a fingerprint. But first, we introduce our experimental setup and the resulting data set in the next section.

4. DATA SET

In this section, we describe our experimental setup. Our goal was to generate a representative data set containing APs from various vendors and a variety of fingerprinter machines. Therefore, we placed twelve different physical APs—2 AVM FritzBox 7050 (AP1–2), 1 LANCOM L-54G (AP3), 1 D-Link DAP-1360 (AP4), 2 Edimax EW-7228APn (AP5–6), 4 Linksys WRT54GL (AP7–10) and 2 Netgear WG602v4 (AP11–12)—in a dedicated room, all operating on the same channel (for simplicity of data collection). We induced different room temperatures by turning the heating off/on or opening/closing the windows from time to time. We measured the room temperature with a Voltcraft DL-181THP data logger. To measure the clock skew of the APs we used four different fingerprinter machines. All were customary laptops running Ubuntu 10.10. Beacons were sniffed using the modified scapy library proposed in [18].

The experiment was performed over about four weeks. The temperature conditions of our experimental environment are shown in Figure 2, and, as depicted cover all temperatures that are typically to be expected in non-airconditioned indoor settings, i. e., between 17–27 °C. We deliberately created periods where the temperature remained

[7] In fact, all clocks considered here, i. e., the TSF clock and the system clock, have this resolution.

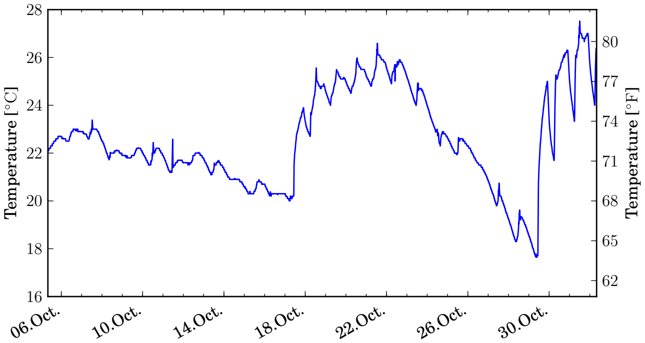

Figure 2: Temperature variation over time

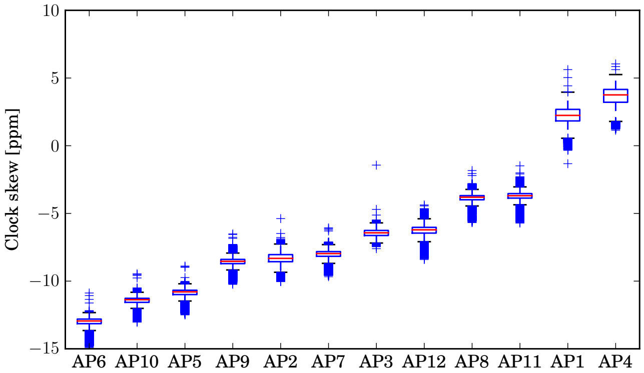

Figure 3: Clock skews (NTP corrected) of the access points in our data set

relatively stable over several days (e. g., weeks 1–2) as well as others with multiple sudden temperature fluctuations (e. g., week 4).

We estimated the clock skew of the twelve APs from the four FP machines for all consecutive intervals with a duration of ten minutes each using LSF linear regression. Note that in practice much shorter intervals are feasible to obtain precise clock skew approximations. With the online algorithm for LSF proposed in [18], 30–80 seconds of beacon sniffing would be enough to obtain a sufficiently small margin of error for the skew approximation. However, we refrain from applying this method to achieve a very high accuracy.

The data logger measured the temperature with a specified accuracy of ±0.1 °C once per minute and we took the mean of the ten measurements as the room temperature for one estimation interval.

The distribution of the APs' clock skews in our sample is shown in the box plots of Figure 3. Here, we eliminated the FP's influence on the measurement using the NTP method (described in more detail in the following section). As we can see, most APs belong to groups of similar clock skews (e. g., AP9, AP2, and AP7 or AP8 and AP11). Hence, even in such a small sample of APs, there is already significant confusion between the APs based on their clock skews. In the following, we will show how to radically improve the uniqueness of clock skew based fingerprints and, hence, the discriminability of APs by considering the dependency of the clock skew on temperature. But before that, we describe how to eliminate the distortion caused by the fingerprinter.

5. FINGERPRINTER INFLUENCE

In this section, we briefly revisit the previously proposed and, up to now, only available method for eliminating the influence of the fingerprinters' own clock skews, i. e., the *NTP method* [18]. It relies on an *estimation of the fingerprinters' skews* from a Network Time Protocol (NTP) service running on the fingerprinters. We show why this approach is insufficient for examining the dependency of clock skews on temperature. We then introduce a novel approach, the *2AP method*, which is based on the observation that the fingerprinters' skews are cancelled when calculating *clock skew differences* between two measured access points. Hence, this method eliminates the fingerprinter influence by design.

The NTP method is based on the observation that, after rewriting equation (2) for the (objective) skews of access point and fingerprinter to

$$C(t_2) - C(t_1) = [1 + \mathrm{s}_C(t_1, t_2)] \cdot (t_2 - t_1) \qquad (5)$$
$$D(t_2) - D(t_1) = [1 + \mathrm{s}_D(t_1, t_2)] \cdot (t_2 - t_1) \qquad (6)$$

and employing this substitution in the equation for subjective skew (4), we obtain:

$$\begin{aligned} \mathrm{s}_{D,C}(t_1, t_2) &= \frac{\mathrm{s}_C(t_1, t_2) - \mathrm{s}_D(t_1, t_2)}{1 + \mathrm{s}_D(t_1, t_2)} \\ &\approx \mathrm{s}_C(t_1, t_2) - \mathrm{s}_D(t_1, t_2) \end{aligned} \qquad (7)$$

The last approximation holds since $\mathrm{s}_D(t_1, t_2)$ is in the range ±30 ppm and, hence, division by $1+\mathrm{s}_D(t_1, t_2)$ will maximally be relevant in the fifth significant digit.

We can rewrite this equation again to

$$\mathrm{s}_C(t_1, t_2) \approx \mathrm{s}_{D,C}(t_1, t_2) + \mathrm{s}_D(t_1, t_2) \qquad (8)$$

and, thereby, have an expression of the (objective) skew of C in terms of the subjective skew of C as seen by D and (an approximation of) the (objective) skew of D. The latter is obtained from an NTP service running for 48 hours on each FP. In the following, we will call all skew approximations obtained according to this method *NTP corrected clock skews*.

If we apply this correction of subjective skews to the example from Figure 1, we obtain the situation shown in Figure 4. We can see that the influence of the FPs' clock skew is significantly reduced, while the APs' clock skew still diverges substantially at different measuring times. The remaining inaccuracies of the fingerprinter skew approximations (especially visible in the slightly different slopes at measuring time m_1) are due to the fact that the results are only corrected by the *average* skew of the fingerprinters. This imprecision is enough to render the NTP method inappropriate for examining temperature dependency, as will be shown below.

Our proposal to overcome this inaccuracy is to calculate the difference between the subjective clock skews of two access points measured at the same time. We utilize the fact that Wi-Fi hotspots are rarely alone in their coverage area. By performing the 2AP method, we completely remove the influence of the fingerprinter. The difference between the subjective skews of AP clocks C and C' as measured by FP clock D is equal to the difference of the corresponding (objective) clock skews (except for practically irrelevant de-

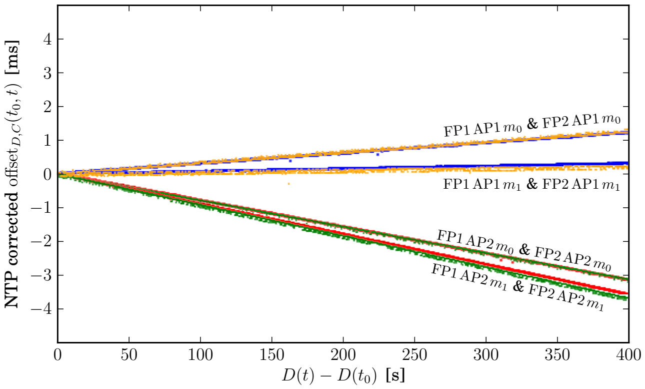

Figure 4: Subjective offsets from Figure 1 corrected by NTP method

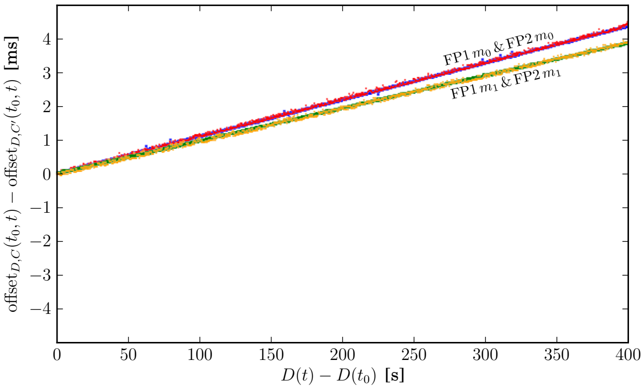

Figure 5: Offset differences of 2AP method (between C = AP1 and C' = AP2) for subjective offsets from Figure 1

viations), similarly to (7):

$$\begin{aligned} \mathrm{s}_{D,C}(t_1,t_2) - \mathrm{s}_{D,C'}(t_1,t_2) &= \frac{\mathrm{s}_C(t_1,t_2) - \mathrm{s}_{C'}(t_1,t_2)}{1 + \mathrm{s}_D(t_1,t_2)} \\ &\approx \mathrm{s}_C(t_1,t_2) - \mathrm{s}_{C'}(t_1,t_2). \end{aligned} \tag{9}$$

Again, the last approximation holds since s_D is in the range ±30 ppm and, hence, changes due to multiplication or division by $1 + \mathrm{s}_D$ can be neglected.[8] In the following, we call all data obtained according to this method *2AP clock skew differences*, where we will without loss of generality always subtract the subjective skew of the AP with the lexicographically greater MAC address from that of the other.

In Figure 5, the offset differences of the clocks from Figure 1 are shown. We observe that taking these differences completely removes the influence of the fingerprinters' *current* skews and, hence, leads to values that are—in contrast to the NTP corrected skew approximations—*fully* comparable between different fingerprinters. The difference between the different measuring times is still visible and our main claim is that it is, to a large extent, due to the temperature dependency of the access points' clocks.

Using the 2AP method, we get meaningful values only for *pairs* of access points. Hence, it is at first ambiguous which of the two APs is responsible for a possible mismatch. Still, the information provided is essential. If more than two access points are visible then several or all possible pairs can be examined and a faked access point should lead to mismatches in significantly more of its pairs than the others. Moreover, the sole presence of a faked access point might be enough to render a whole environment untrustworthy or at least suspicious.

In Figure 6, we show samples of clock skew measurements as a function of temperature for two APs with the NTP method and the corresponding pair for the 2AP method, i. e., the approximate skews and skew differences in parts per million (ppm) of our ten minute measuring intervals are plotted over the average temperatures in these intervals. Apparently, the FPs have very distinct dependencies on temperature leading to different distributions of the data for the NTP method. These dependencies are intrinsic and cannot be modeled independently of the fingerprinter. Besides the variations caused by the FPs, the intrinsic temperature dependency characteristics of the APs are also clearly visible. As described above, for the 2AP method, only the APs' dependency on temperature remains, while the FPs' dependency is completely eliminated. Therefore, the 2AP method yields a fully fingerprinter-independent estimation of the clock skew differences and can be used to further examine the temperature dependency of the APs' clocks.

Accordingly, in the following section, we show how to model the temperature dependency for the 2AP method in order to drastically increase the fingerprinting accuracy. Finally, we will analyze the fake AP detection efficiency of our methods.

6. TEMPERATURE DEPENDENCY

Theoretically, common crystal oscillators should exhibit a cubic dependency on temperature (see Section 3.1). However, our practical observations do not confirm this assumption. There are several possible reasons: Firstly, the cubic dependency is to be expected for crystal oscillators against the temperature of the crystal, while we measure the dependency of TSF clock skews on room temperatures. The dependency between room temperature and crystal temperature may be non-trivial, e. g., due to different (passive) cooling behavior of the AP. Besides, the regarded temperatures might cover only a small section of that cubic function. Secondly, there might be other physical influences that correlate with temperature and change the frequency and, hence, the skew. Thirdly, the crystals used in certain APs might not be AT-cut crystals that exhibit this cubic dependency (or suboptimal quality of the cuts might lead to a deviation from the theoretical properties).

Therefore, we use *Gaussian process regression (GPR)* [22], a method that does not make any assumptions about the underlying function and thus is suitable as a generic approach for modeling the dependency. The goal of our model is to predict values $f(x_*)$ of the clock skew difference of two APs that is to be expected at a given room temperature x_*. In general, a Gaussian process can represent $f(x)$ indirectly by considering the observed data. Each observation

[8] Observe that this is, by equation (7), also approximately equal to the subjective skew $\mathrm{s}_{C',C}(t_1,t_2)$ of access point C as seen by *access point C'*. Similar observations concerning clock skew arithmetic have been already made by Arackaparambil et al. [1], but not applied to eliminate fingerprinter influence.

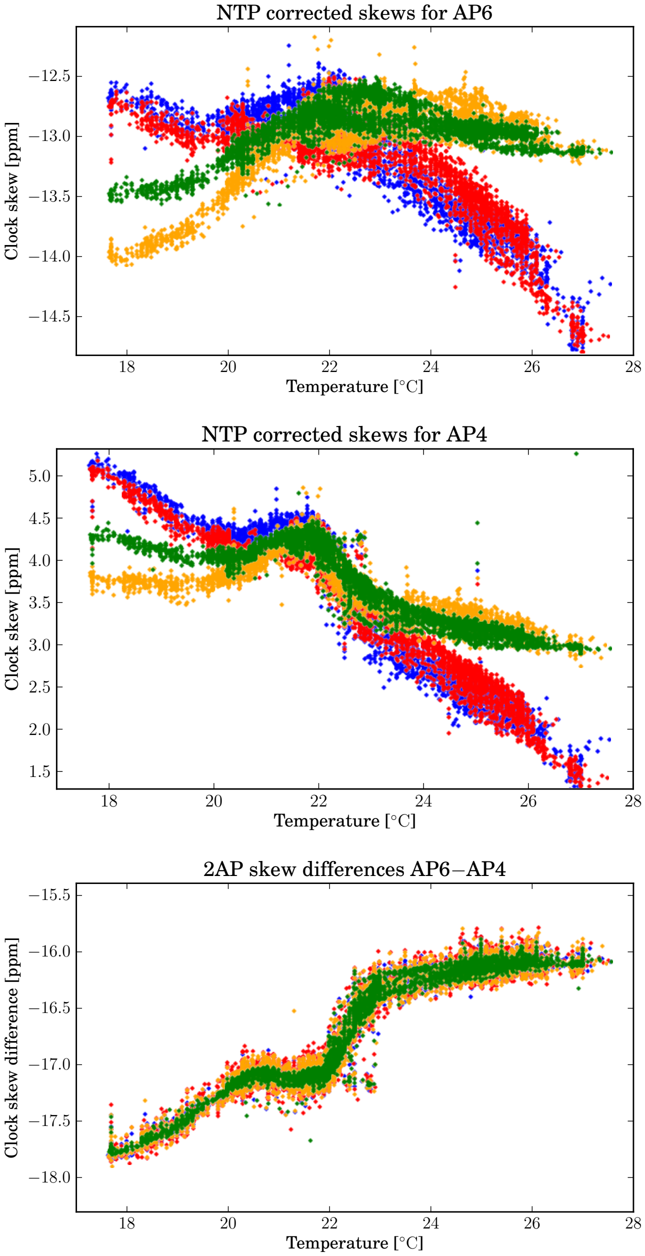

Figure 6: NTP-corrected clock skews and 2AP clock skew differences in relation to room temperature measured by four FPs (different colors)

$y_i \in \{y_1, ..., y_n\}$ for an input x_i is assumed to be a single point sampled from a multivariate Gaussian distribution, given by $y_i = f(x_i) + \varepsilon$, where $\varepsilon \sim \mathrm{N}(0, \sigma_n^2)$ is the additive noise term. A Gaussian process, specified by its mean function $m(x)$ (usually assumed to be zero) and the covariance function $k(x_p, x_q)$ can be used to define a prior over possible functions, $f \sim \mathrm{GP}(m(x), k(x_p, x_q))$. For the posterior distribution, the prior is restricted to contain only functions that meet the observed data. This is done by computing for each test input x_* a predictive mean $f(x_*)$, which is a linear combination of *all* training inputs and outputs built according to the covariance function, and a predictive variance $\mathbb{V}[f(x_*)]$, which only depends on the training inputs. GPR is designed to interpolate predictions for unobserved inputs. In practice, we have precomputed and stored the predictive mean and predictive variance for all temperatures needed in our evaluation with a granularity of 0.1 °C.

In our model, we use the *squared-exponential covariance function*

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x_p - x_q)^2\right) + \sigma_n^2 \delta_{pq}$$

as it meets our requirements well: values whose inputs are close get a higher influence on the prediction, while distant observations have negligible effect; it is also infinitely differentiable, leading to smoothness of the generated predictions. It is parameterized by *hyperparameters* $\theta = (\sigma_f^2, \ell, \sigma_n^2)$, where σ_f^2 is the variance of the signal itself, ℓ is the length-scale and σ_n^2 is the expected variance of measurement noise. We estimate these hyperparameters by optimizing the marginal likelihood on our training data. For a detailed description we refer to Rasmussen and Williams [22].

Figure 7 shows examples for GPR applied to our data. We can see that the predicted mean functions perfectly fit our observed data. We verified that comparable predictions can be obtained using much smaller samples of about 100 observations.

When using clock skews (or clock skew differences) as fingerprints, suitable intervals for recognition (as shown in Figure 7) have to be defined due to the volatile nature of these measures. In the following section, we provide a formal definition of such recognition intervals.

7. EVALUATION OF PREDICTION

In this section, we first define recognition methods based on the results described above. We then evaluate the accuracy of these methods regarding fake AP detection. Finally, we provide an information theoretical perspective on temperature dependency as a feature for fingerprinting.

7.1 Recognition Intervals

In order to provide recognition methods for APs, we specify appropriate acceptance intervals of observed data for all considered methods. Regardless of the method, we classify 5% of our training data as outliers.

Therefore, we define this interval for the corrected skews of the NTP method and for the skew differences of the 2AP method as $\mu \pm 2\sigma$, where μ is the mean and σ the standard deviation of a normal distribution fitted to the training data. The NTP method directly allows the recognition of an AP, while the 2AP method needs at least one additional AP for comparison (as described in Section 5).

To apply temperature dependent skew difference predictions, we derive a new method, called *2AP-T*, that employs Gaussian process regression (as described in the previous section). Here, the recognition interval is specified by $f(x_*) \pm 2\sqrt{\mathbb{V}[f(x_*)]}$, where $f(x_*)$ is the predicted mean and $\mathbb{V}[f(x_*)]$ the predicted variance of the corresponding GPR, given the current room temperature x_*.

The properties of normal distribution and GPR ensure that the recognition intervals comprise about 95% of the training data.

To evaluate the discriminability of fingerprints generated by these three methods, we calculated their average recog-

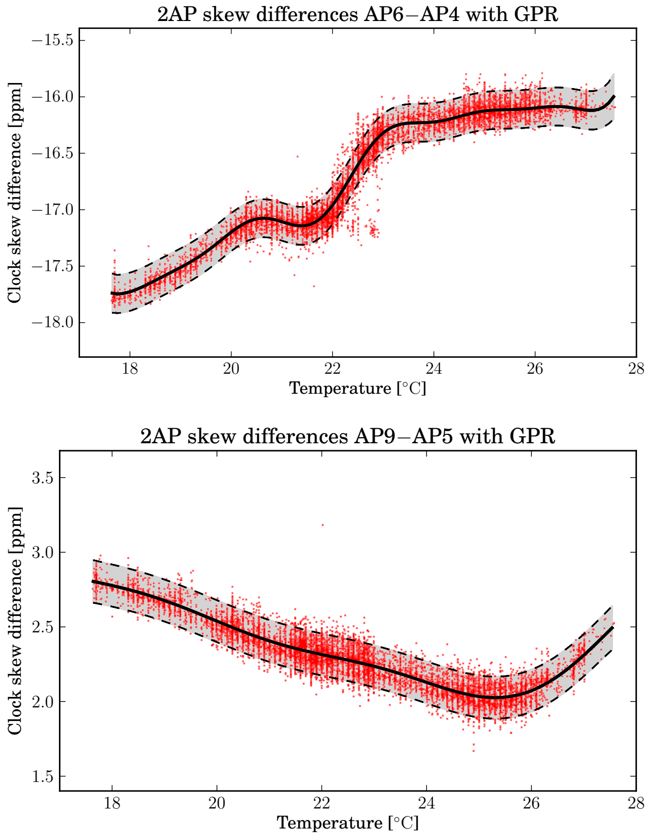

Figure 7: Two examples for GPR: pointwise mean prediction $f(x)$ $(\pm 2\sqrt{\mathbb{V}[f(x)]}$, corresponding to the 95 %-confidence region)

nition interval size. We obtained 1.73 ppm for the NTP method, 1.56 ppm for the 2AP method and 0.41 ppm for the 2AP-T method. The results reveal that the temperature-dependent prediction of the 2AP-T method improves fingerprinting discriminability by more than 75% on average compared to NTP. In the following, we examine how the three methods perform in detecting whether an AP has been replaced by an attacker.

7.2 Detection of Fake APs

We consider two attacker types: The *naïve attacker* has no information on the original AP's clock skew. He randomly selects an available AP for the attack. Practically, we model this attacker by replacing a considered AP with every other AP in our data set and taking the mean success rate. The *intelligent attacker* knows about the clock skews and the detection method used. Thus, he is able to select the AP with the clock skew closest to the original AP's clock skew in our data set as the replacement. Recall that we do not consider attackers who perform extensive hardware modifications to control the clock skew.

We divided our data into training and testing as follows: all data was separated into four equally long time periods (corresponding to approximately one week each). Training was performed by three (out of four) fingerprinters on the data of three (out of four) weeks and tested by the remaining fingerprinter on the data from the remaining week. To get a representative result, we performed 16-fold cross validation for all possible combinations. The separation of data was deliberately not random but followed a logical distinction. This guaranteed that the testing FP was never used for training and samples in the testing period had a significant temporal distance from those in the training period. This corresponds to a scenario where several users created fingerprints over time and a new user compares his current fingerpint against these at some different point in time, as exploited in our proposed architecture (Section 9).

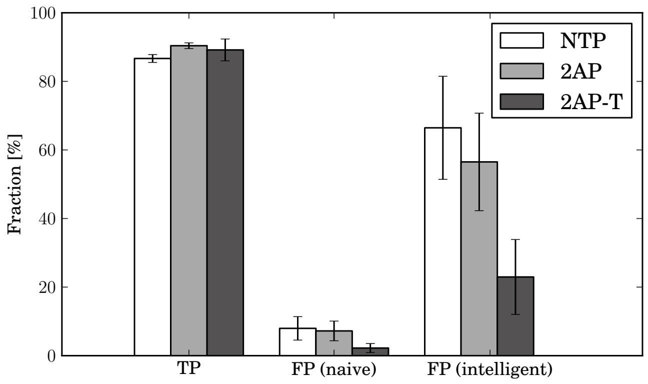

Figure 8: True positive (TP) and false positive (FP) rates for different attacker models; error bars show variation (95% confidence) over different original APs

Our evaluation focuses on two measures: whether an original AP is recognized (true positive) and whether a faked AP succeeds in spoofing the identity (false positive). Figure 8 shows the result. The true positive rate (TP) for 2AP-based methods is marginally better than for NTP and all rates are around 90%. This is slightly less than the targeted 95%, as the acceptance intervals are generated on disjoint training data. More interesting are the results for false positives (FP). A naïve attacker has hardly a chance of succeeding: All of our methods detect the impostor in more than 90% of all cases. The best results are obtained with the 2AP-T method (FP rate 2.23%).

The effectiveness of our method is particularly impressive when dealing with an intelligent attacker. While with the NTP method the attacker succeeds in 67% of all cases, the 2AP method decreases his success rate to 56% and the 2AP-T method to 22%. Regardless of the attacker type, the 2AP-T method is able to improve the spoof detection rate by a factor of three. Since the false positive rates for the intelligent attacker exhibit considerable variations for different original APs (indicated by error bars in Figure 8), we show the separated false positive rates per orginal AP and method in Figure 9. As we can see, in general the 2AP method outperforms NTP. However, the 2AP-T method is always significantly better than both the NTP and the 2AP methods. The error bars for the 2AP and 2AP-T methods indicate the variance over the different comparison APs. The 2AP method is prone to large variations caused by the different temperature dependency characteristics of the comparison APs. This effect is drastically reduced by 2AP-T, which explicitly considers the current temperature. We further see, that even for APs with similar attacker success rates against NTP and 2AP (e. g., AP5/7 or AP2/8), the success rate against the 2AP-T method differs notably. 2AP-T per-

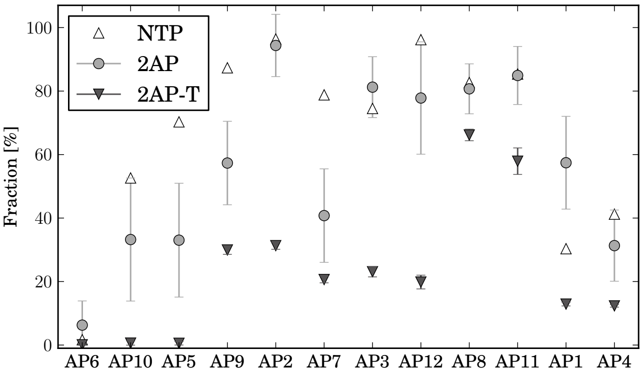

Figure 9: FP rate for intelligent attacker per AP; error bars show variation over comparison APs for 2AP and 2AP-T (95% confidence)

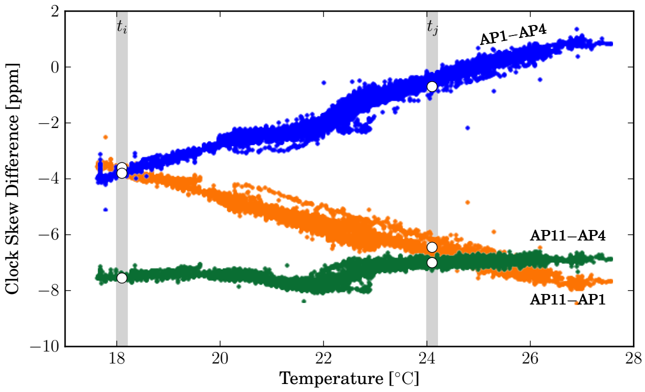

Figure 10: Exemplary input for the SVM

forms best for cases where temperature dependency is very pronounced.

We now present an information theoretical perspective on how much information the temperature contributes to the classification of the APs in our sample.

7.3 Information Theory

From an information theoretical point of view, we are interested in how much information is contained in the temperature dependency, which we use as additional feature for the 2AP-T recognition/fake detection method. A common evaluation measure is the *mutual information (MI)*, defined for two random variables X and Y as $I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)\,p(y)}$. In our case, Y represents the considered AP pair, while X denotes the classification feature, i. e., clock skew difference for the 2AP method and the combination of clock skew difference and temperature for the 2AP-T method.

The concept of MI is closely related to entropy and results are expressed as bits. For feature selection, MI measures how much information the presence of a feature contributes to making the correct classification decision. In the case of our methods, it measures how much additional information is provided by the temperature dependency (2AP-T) compared to observing only the clock skew difference (2AP). For details about the calculation we refer to [19]. For our data set, we obtain an MI for 2AP of 4.12 bits and for 2AP-T of 5.16 bits. Note that for our data, the MI is upper bounded by 6.04 bits (as we are classifying 66 different pairs). Therefore, the (information theoretically) perfect feature for classification cannot contribute more than 6.04 bits of information. We conclude that the knowledge of temperature dependency contains more than half of the remaining uncertainty of access point identification when combined with 2AP clock skew differences.

8. METHOD WITHOUT EXPLICIT KNOWLEDGE OF TEMPERATURE

Although our approach provides reliable detection of replaced APs, it depends on information that is not always available in measuring devices: the temperature that APs are exposed to. The question we address in this section is how, if at all, we can perform the detection of impostors if no explicit temperature information is available.

We assume that there are at least three APs transmitting in the environment to be evaluated (say, these are AP1, AP4, and AP11). For every time instance, the fingerprinter is able to derive three 2AP clock skew differences: AP1−AP4, AP11−AP4, AP11−A1. The key idea of our approach is to learn legitimate combinations of simultaneously occurring differences. Figure 10 shows an example of two time instances, time t_i with clock skew differences $(-3.797, -3.607, -7.552)$ and time t_j with $(-0.704, -6.451, -7.013)$. We use such triples of simultaneously measured clock skew differences as vectors to train a machine learning technique. Three APs is the minimum number required because from two APs only one difference could be derived. Note that when using only two APs the machine learning approach is equivalent to the 2AP method and does not learn legitimate combinations but rather single differences. The proposed method only detects whether one of the three APs is faked and not which one. This is similar to the 2AP and 2AP-T methods only detecting whether one out of two APs is potentially spoofed. As described in Section 5, a greater number of reachable APs can be used to identify the actual fake AP.

We apply support vector machines (SVMs), state-of-the-art classification methods used in machine learning, which are well-known for their high performance in terms of classification accuracy. The technique dates back to the work of Vapnik and Chervonenkis [28] in 1974. The focal idea is the interpretation of instances as vectors in a vector space. Based on training data, the classifier tries to fit a hyperplane into the vector space which separates the instances that belong to different classes. In our case (one class SVM) the plane is fitted in such a way that the training data is separated from the origin whereby a fraction of at most ν (which is a SVM parameter) training points are allowed to be outside the estimated region.

The plane is fitted such that the accumulated distance between the closest instances (support vectors) and the plane is maximized to ensure a clear distinction between the classes. In cases where the vectors are not linearly separable, the vector space is transformed into a higher dimensional space where the linear separation is possible (the *kernel trick*). An interested reader is pointed to [5] for thorough information about SVMs.

Sample Size [% / #]	**TP [%]** **avg/mdn**	**FP [%]** **avg/mdn**
100 / 3520	87.26 / 96.88	12.18 / 0.00
0.05 / 300–400	85.95 / 96.34	11.07 / 0.00
0.015 / 130–160	84.53 / 94.81	10.96 / 0.00
0.005 / 40–50	80.48 / 90.13	10.54 / 0.00

Table 1: True and false positive rate without explicit temperature knowledge

We divided our data into training and testing in the same way as in previous sections. The parameter ν defines an upper bound on the fraction of outliers and, at the same time, a lower bound on the fraction of support vectors (i. e., the generalizability of the model). We set $\nu = 0.05$; hence, at most 5% of training data may be assigned to false negatives by the model. Table 1 (first row) shows the results of the evaluation. We choose the intelligent attacker model described above (i. e., he replaces the original AP with a fake AP that best matches the original clock skew). We skipped the cases when the best fake AP is already included in the triple. As our evaluation shows, on average more than 87% of trustworthy environments are recognized as such (true positive). The median reaches almost 97%. The intelligent attacker is successful in only about 12% of all cases (false positive).

The result is at first glance superior to the 2AP-T method. However, the two methods cannot be directly compared as they consider different numbers of simultaneously reachable APs.

Nevertheless, the results without explicit knowledge of temperature provide surprisingly high accuracy. This is achieved even without tuning SVM parameters. We assume that by optimizing them, one would get even better results (but we omit this due to the high accuracy classification even with the default parameters).

Recall that all our observations were collected in an environment where all APs shared the same temperature exposure. In practical applications, this might not hold true. However, we expect our method to still provide significant detection accuracy as long as the different temperatures are correlated, e. g., due to outdoor temperature or time of day.

For practical relevance, it is important to know how many observations (training data) are needed to achieve good classification accuracy. To determine this number, we performed our evaluation by randomly selecting only 0.05%, 0.015%, and 0.005% of all available training data. This corresponds to roughly 350, 150, and 50 samples. The results are shown in Table 1 in rows two to four. As expected, less training data leads to lower classification accuracy. However, the accuracy degradation is very slight: using only 50 training samples, on average the accuracy is as high as 80% for true positives and 10% for false positives. These results underline the practical relevance of our method as only a few dozen observations without any temperature information are sufficient to learn the parameters of a trustworthy environment.

9. ARCHITECTURE

To exploit the results described above, we propose an architecture based on a crowdsourcing approach (Figure 11). The core of this system is a *trusted service* (TS) that collects fingerprints, performs the necessary calculations and provides feedback to users. Assume a user who wants to connect to a (potentially untrustworthy) access point (UAP) and wants to ensure its trustworthiness. A client application (*app*) first extracts timing information from beacon frames of all receivable APs, calculates the respective clock skews and, if available, measures the current temperature. The pairs of MAC address and clock skew with the optional temperature information represent the *fingerprints*. A secured channel to the TS can be established in two possible ways: 1. If the UAP provides free access to the Internet, using its connectivity, 2. If not possible (e. g., because access is bound to providing credit card information before enabled) via a side channel, e. g., 3G. Encryption and authenticity can be ensured by using well-known certificate-based standards such as SSL. Note that a man-in-the-middle attack on this channel by the UAP can be mitigated by hard-coding the certificate into the app. The client then sends the fingerprint via the secured channel to the TS. The TS decides, depending on the information provided by the client, which of the different proposed methods for verification (SVM/2AP-T/2AP) to use and calculates its decision about the trustworthiness (the *verdict*)—either binary or score-based. The verdict is sent back to the client, enabling the user to decide whether to use the UAP or not. All fingerprints queried for verification are stored and integrated by the TS into its assessment. The more users use the system, the broader is the base for this assessment and, therefore, its precision. Thereby, APs establish a reputation.

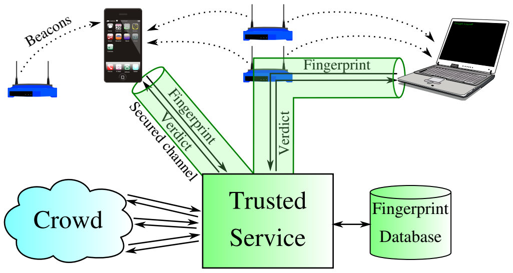

Figure 11: The architecture

A legitimate replacement of an AP by the operator will not cause any confusion: the new device will have a different MAC address and, hence, will be recognized as new AP in that environment. Therefore, our crowdsourced approach will automatically integrate this AP and begin to build its reputation. Note that an attacker would also be able to integrate his AP in this way. However, this would require him to provide reliable long-term service to gain sufficient reputation. The described approach could be implemented without changing any standardized protocols and without requiring the cooperation of network operators—instead, the incentive for using this system is shifted to users who care about their security. In future work we plan to implement this architecture and test its effectiveness in a real-world scenario.

Finally, it would also be possible to store the trained models of the SVM for a favored set of APs in the app itself, as the verification of an environment against a trained SVM model requires only low computational resources and the models need only small storage capacities (the trained models in our experiments are around 2–32KB in size).

10. CONCLUSION

In this paper, we provide a practical solution to reliably detect faked access points. This is done by passively estimating the clock skew from information contained in management frames. We show a way to significantly increase its information content by considering its dependency on temperature. Additionally, our method completely eliminates the influence of the measuring device on the fingerprint. Hence, measurements performed by different clients become comparable. Interestingly, our method works even without explicit access to temperature information and protects even against an attacker who selects a fake AP with a similar average clock skew.

Our method exploits an intrinsic physical property that provides the highest discriminability known so far of APs using standard client hardware. The effort for simulating this property is much more complex than adjusting the average clock skew, even if hardware is modified. This renders the attack practically infeasible.

Our approach yields a strong feature for passive remote physical device fingerprinting in wireless networks. Using only 50 observations for training, our approach detects faked access points in 90% of all cases.

No currently deployed protection mechanism for public hotspots (e. g., web-based authentication) provides any security against the described threat. With our proposed architecture it is possible to mitigate the danger without changing deployed systems or the standard protocols used.

Acknowledgements

The authors would like to thank Andreas Zinnen for valuable discussions about machine learning techniques.

This work has been supported by the National Research Fund of Luxembourg (FNR), the FNR CORE project MOVE, and the EU FP7 PPP project CONFINE.

11. REFERENCES

[1] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz. On the reliability of wireless fingerprinting using clock skews. In *3rd ACM Conference on Wireless Network Security (WiSec '10)*, pages 169–174. ACM, 2010. http://dx.doi.org/10.1145/1741866.1741894.

[2] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. Enhancing the security of corporate wi-fi networks using DAIR. In *4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, pages 1–14. ACM, 2006. http://dx.doi.org/10.1145/1134680.1134682.

[3] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles. Active behavioral fingerprinting of wireless devices. In *1st ACM Conference on Wireless Network Security (WiSec '08)*, pages 56–61. ACM, 2008. http://dx.doi.org/10.1145/1352533.1352543.

[4] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pages 116–127. ACM, 2008. http://dx.doi.org/10.1145/1409944.1409959.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. http://www.support-vector.net/.

[6] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, pages 581–590. ACM, 2006. http://dx.doi.org/10.1145/1124772.1124861.

[7] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *15th USENIX Security Symposium (SSYM '06)*, pages 167–178. USENIX Association, 2006. https://www.usenix.org/legacy/events/sec06/tech/full_papers/franklin/franklin.pdf.

[8] K. Gao, C. Corbett, and R. Beyah. A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '10)*, pages 383–392. IEEE Computer Society, 2010. http://dx.doi.org/10.1109/DSN.2010.5544294.

[9] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy, and D. Sicker. Practical Defenses for Evil Twin Attacks in 802.11. In *IEEE Globecom Communications and Information Security Symposium (Globecom 2010)*, Miami, FL, December 2010.

[10] J. Hall, M. Barbeau, and E. Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. In *3rd IASTED International Conference on Wireless and Optical Communications (WOC '03)*, pages 13–18. ACTA Press, 2003. http://people.scs.carleton.ca/~kranakis/Papers/RFFPaper3.pdf.

[11] IEEE Computer Society. *Standard 1193-2003: IEEE Guide for Measurement of Environmental Sensitivities of Standard Frequency Generators.* http://standards.ieee.org/findstds/standard/1193-2003.html.

[12] IEEE Computer Society. *Standard 802.11-2012: IEEE Standard for Information technology – Telecommunications and information exchange between systems, Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* http://standards.ieee.org/findstds/standard/802.11-2012.html.

[13] IETF. *RFC 1323: TCP Extensions for High Performance*, May 1992. http://www.rfc-editor.org/rfc/rfc1323.txt.

[14] IETF. *RFC 2865: Remote Authentication Dial In User Service (RADIUS)*, June 2000. http://www.rfc-editor.org/rfc/rfc2865.txt.

[15] S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. In *14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pages 104–115. ACM, 2008. http://dx.doi.org/10.1145/1409944.1409958.

[16] Jauch Quartz GmbH. *Quartz Crystal Theory*, 2007. http://www.jauch.de/ablage/med_00000619_1193753698_Quartz%20Crystal%20Theory%202007.pdf.

[17] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *IEEE Transactions on*

Dependable and Secure Computing, 2(2):93–108, 2005. http://dx.doi.org/10.1109/TDSC.2005.26.

[18] F. Lanze, A. Panchenko, B. Braatz, and A. Zinnen. Clock skew based remote device fingerprinting demystified. In *2012 IEEE Global Telecommunications Conference (GLOBECOM '12)*, pages 813–819. IEEE Computer Society, 2012. http://dx.doi.org/10.1109/GLOCOM.2012.6503213.

[19] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. http://nlp.stanford.edu/IR-book/.

[20] S. B. Moon, P. Skelly, and D. F. Towsley. Estimation and removal of clock skew from network delay measurements. In *18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, pages 227–234. IEEE Computer Society, 1999. http://dx.doi.org/10.1109/INFCOM.1999.749287.

[21] C. Neumann, O. Heen, and S. Onno. An empirical study of passive 802.11 device fingerprinting. In *32nd International Conference on Distributed Computing Systems Workshops (ICDCSW '12), Workshop on Network Forensics, Security and Privacy (NFSP)*, pages 593–602. IEEE Computer Society, 2012. http://dx.doi.org/10.1109/ICDCSW.2012.8.

[22] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. http://www.gaussianprocess.org/gpml/.

[23] V. Roth, W. Polak, E. Rieffel, and T. Turner. Simple and effective defense against evil twin access points. In *1st ACM Conference on Wireless Network Security (WiSec '08)*, pages 220–235. ACM, 2008. http://dx.doi.org/10.1145/1352533.1352569.

[24] D. Shaw and W. Kinsner. Multifractal modelling of radio transmitter transients for classification. In *Communications, Power and Computing (WESCANEX '97)*, pages 306–312. IEEE Computer Society, 1997. http://dx.doi.org/10.1109/WESCAN.1997.627159.

[25] B. Sieka. Active fingerprinting of 802.11 devices by timing analysis. In *3rd IEEE Consumer Communications and Networking Conference (CCNC '06)*, pages 15–19. IEEE Computer Society, 2006. http://dx.doi.org/10.1109/CCNC.2006.1592979.

[26] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of SSL warning effectiveness. In *18th USENIX Security Symposium (SSYM '09)*, pages 399–416. USENIX Association, 2009. https://www.usenix.org/legacy/event/sec09/tech/full_papers/sunshine.pdf.

[27] M. B. Uddin and C. Castelluccia. Toward clock skew based wireless sensor node services. In *5th Annual ICST Wireless Internet Conference (WICON '10)*, pages 1–9. IEEE Computer Society, 2010. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5452689.

[28] V. N. Vapnik and A. J. Červonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, 1974. (German Translation: Theorie der Zeichenerkennung, Akademie-Verlag, 1979).

[29] J. R. Vig. Introduction to quartz frequency standards. Research and Development Technical Report SLCET-TR-92-1, U. S. Army Electronics and Devices Laboratory, March 1992. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA248503.

[30] Z. Yang, L. Cai, Y. Liu, and J. Pan. Environment-aware clock skew estimation and synchronization for wireless sensor networks. In A. G. Greenberg and K. Sohraby, editors, *INFOCOM 2012*, pages 1017–1025. IEEE, 2012.

[31] H. Zhou, C. Nicholls, T. Kunz, and H. Schwartz. Frequency accuracy & stability dependencies of crystal oscillators. Technical Report SCE-08-12, Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, November 2008. http://kunz-pc.sce.carleton.ca/thesis/CrystalOscillators.pdf.