2016

Device fingerprinting identification and authentication: A two-fold use in multi-factor access control schemes

Paul Eugene Manning
Iowa State University

Device fingerprinting identification and authentication: A two-fold use in multi-factor access control schemes

by

Paul E Manning

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Information Assurance

Program of Study Committee:
Doug W. Jacobson, Major Professor
Akhilesh Tyagi
George Amariucai

Iowa State University

Ames, Iowa

2016

DEDICATION

My wife and family have been very supportive of my efforts to obtain my degree. Every step of the way my wife made certain I had the space and the time to complete all the course work and read many of my papers and sometimes to check my math. I also wish to dedicate this to my parents who on frequent telephone calls would always ask, “Are you still going to the University?” and “When do you finish?” As always, they kept a lamp on so I could find my way home. I would be remiss if I did not mention two very important people at Iowa State who helped make this experience productive and made certain that even at a distance I could accomplish everything necessary, Virginia Anderson, for your guidance and doing the legwork for all those administrative tasks and Dr. Doug W. Jacobson both who took a telephone call in 2012 and turned a question into a life changing event. Finally, I want to mention my committee for their support, their answering my questions, and their guidance as I worked through the issues. To the above people I wish to dedicate this thesis and hope it is worthy of their efforts on getting me to this place in my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

AP	Access Point
EAP	Extensible Authentication Protocol
EU	European Union
ICMP	Internet Message Control Protocol
IEEE	Institute of Electrical and Electronic Engineers
IoT	Internet of Things
LAN	Local Area Network
NAC	Network Access Control
NAT	Network Address Translation
RFC	Request for Comment
TCP	Transmission Control Protocol
WLAN	Wireless Local Area Network

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Doug Jacobson, and my committee members, Dr. George Amariucai, and Dr. Akhilesh Tyagi, for their guidance and support throughout the course of this research. I also mention Dr. Tom Daniels who shortened my anxiety by informing me the research had been done so review and point out what supports my thesis as well as reminding me a picture is worth a thousand words. I also want to mention the students of many of my classes whose comments, presentations, and feedback was inspiring, while their names did not stick many of their comments did. I want to also acknowledge the teaching assistants who saw me through technical glitches, answered my questions, and provided direction when things didn't go as described.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

ABSTRACT

Network security has always had an issue with secure authentication and identification. In the current mixed device network of today, the number of nodes on a network has expanded but these nodes are often unmanaged from a network security perspective. The solution proposed requires a paradigm shift, a recognition of what has already happened, identity is for sale across the internet. That identity is the users' network ID, their behavior, and even their behavior in using the networks. Secondly a majority of the devices on the Internet have been fingerprinted. Use of device fingerprinting can help secure a network if properly understood and properly executed. The research into this area suggests a solution. Which is the use of device fingerprints including clock skews to identify the devices and a dual- authentication process targeted at authenticating the device and the user. Not only authenticating the identity presented but also combining them into a unified entity so failure to authenticate part of the entity means the whole is denied access to the network and its resources.

CHAPTER I

INTRODUCTION

Information networks are not a new creation. They have existed in a variety of forms using signals to communicate information to their current form, computer networks. Security of information networks has been a major concern for as long as men have built information networks. This security has been addressed in a variety of ways through encryption, user identification, the use of wax stamps to identify the sender, all the way to the present with complex multi-factor authentication systems. Most network access solutions work via the authentication of the user with the assumption the machine the user is employing is a valid device and ignoring the machines on the network that are considered unmanageable. These unmanageable machines often do not require a user to login but still pass traffic and data across the pipes [69]. This paper will explore the proposition, device finger printing can be used as part of a multi-factor identification/authentication scheme that adds security to a network. More than just using device fingerprinting as part of the authentication process, I advance a rather novel concept, the use of the device fingerprint as an on ramp to cyberspace. To those two ends this paper will cover its material in the following fashion, an introduction to define the problem, a section where the definition of a key concept, identification and authentication; in information assurance is discussed and refined. Next, having refined the definitions I will look at the current state of device finger printing across the Internet. The next section of the paper, will review the authentication process and look at the Extensible Authentication Protocol paying attention to the version with Transport Layer Security, EAP-TLS. The following section, Chapter Four, will propose a different solution incorporating

device fingerprinting, EAP-TLS, and also propose a paradigm shift regarding computers and networks. In this section the question of is it technically possible will be answered using research from the previous two sections. Also, the question will be answered, what changes will be required to make the paradigm shift. Finally, the question of is it legally possible will be touched on. Any change in user behavior and this would require a change must first satisfy the simple binary algorithm $A = (C_m C_p C_i)$ where A is action for change, C is the user, m is motivation, p is perception, and i is the information. If any of these three factors is 0 then no action for change will occur. It is in this last chapter I hope to satisfy all the above requirements. The research as to whether this is possible has been accomplished and reviewed in the previous two sections so the appendix will contain a link to the various data sets used in the research allowing the reader the opportunity to check data supporting the proposal.

1.1 **Composition of contemporary networks**

A contemporary network may be of mono or poly device composition. In a mono-device network much like many home networks, you have a variety of Wi-Fi devices all connected using the common router or a set of routers with extenders to cover a limited physical area and provide an on-ramp to the world-wide web and its resources. In a more corporate environment the type of devices will shift from a mono-device network to a more mixed environment referred here as a poly device network. These networks will be a mix of Wi-Fi devices and wired devices. In group of Wi-Fi devices, there may also be variety of devices, ranging from a desktop PC with a wireless card in it to mobile devices such as a cellphone or a tablet. Even among the wired devices there can be a variety of different operating systems

(OS), browsers, model numbers, and vendors. As the movement to Bring Your Own Device (BYOD) gains ground the flavor and variety of devices in such a network will make the poly device network environment even more complex. In both environments security and the presence of unauthorized devices becomes an issue.

In the mono device environment, use of resources can be consumed and network throughput for legitimate users are reduced with unauthorized use of Access Points (AP) [46]. In a poly device environment, the number of attack vectors, assuming each device provides a vector, multiplies greatly. Often in such complex environments, the issue of what devices are legitimately on the network becomes an issue [47]. Many approaches to the identifying legitimate devices have been applied and are currently being marketed. In part, this is due to the weakness in the current identification and authentication schema, username and password, practiced by many networks. Though many of the schemas advanced to enhance network access control (NAC) work very well they do not go far enough to ensure that unauthorized or unidentified devices are on the network. In some cases, the NAC has a class of devices it is unable to identify or manage. To this end, I want to advance the use of device fingerprinting as a certificate not only for identification purposes in the authentication process but as part of an ongoing process to ensure only proper authenticated devices are part of the network. The Internet is exploding with the addition of sensors, home appliances, and multiple devices for users ranging from traditional workstations/laptops to personal health devices with their associated controllers/reporters. Each of these items bring their own set of risks, attack vectors, and information data sets. In the connected world of the internet, the data produced, used, and processed by the extensive collection of devices is at risk of hijack, ransom at best and at worse manipulation skewing results in favor of parties with vested

interest in the results. In this environment, it becomes critical to ensure the devices connected to a network have a legitimate reason for being attached. Device fingerprinting offers not only the ability to identify specific devices but used as part of the authentication process can ensure only devices with a legitimate purpose are attached to a network. However, there are many issues that cloud or confuse the discussion of the authentication.

1.2 **Current view of identification and authentication an issue**

One of the main issues with the discussion of authentication is the use of the terms identification and authentication interchangeably [90]. This substituting use creates a problem as the concepts while complementary are not inter-changeable. The authentication is a process in which identification is a sub-process. IEEE Standard fort Local and metropolitan area networks - Port-Based Network Access Control, IEEE 802.1X-2010 provides guidance for "providing compatible authentication, authorization, and cryptographic key agreement mechanisms to support secure communication between devices connected by IEEE 802® Local Area Networks (LANs), this standard:

a) Specifies a general method for provision of port-based network access control.
b) Specifies protocols that establish secure associations for IEEE Std 802.1AE™ MAC Security.
c) Facilitates the use of industry standard authentication and authorization protocols." [42]

For clarity, we can turn to the IEEE Online Dictionary which states in this document, 802.1, authentication is "The process of verifying an identity claimed by or for a system entity. Defined in RFC 4949" [IEEE Standards Online Dictionary, accessed 6 Oct 2016]. In the

same RFC 4949, identification is “an act or process that presents an identifier to a system so that the system can recognize a system entity and distinguish it from other entities.” As you can see from the definitions provided, identification and authentication are not interchangeable rather identification is a sub process within the authentication process. Figure 1 below provides a simplified diagram of the process.

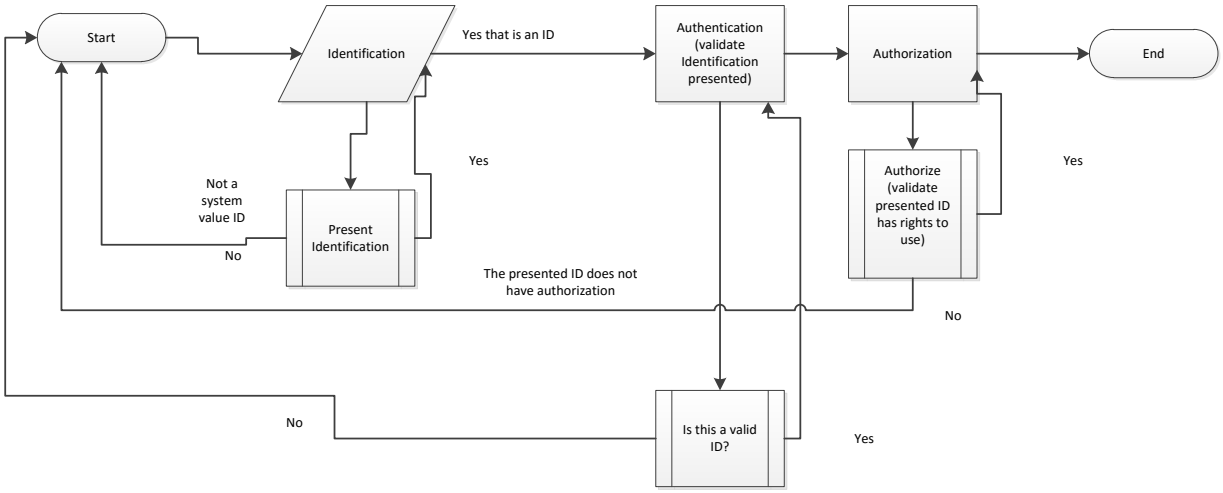

Figure 1 - Authentication Process

An analogy would be the presentation of a driver’s license for identification and the official taking your license (token) and validating it through a series of tests, visual inspection, followed up by calling in via radio or onboard computer, to see if the identification provided matches what is stored in the state’s databases. In the most common schema of identification and authentication, the username often functions as the identity being provided to the system. The password is the verification or authentication that the identity being provided is who they claim to be.

Authorization is another word used as a substitute for the process of authentication [90]. However, this too is an incorrect usage of terminology. In RFC 4949, authorization is

defined as “either an approval that is granted to a system entity to access system resources or a process for granting approval to a system entity to access a system resource.” In calling attention to the phrase “access a system resource,” I am pointing out that this differs from verifying the identity of a system entity and so to use authorization as synonymous with authentication or identification would be incorrect. An incorrect terminology will often lead to incorrect problem definition and may even mask the true problems in a process. By providing this clarification of terminology upfront, I hope to become a little more precise in a probable solution while at the same time provide a perceptual lens to look at the research in this area. Having clarified the definition of the three key terms, identification, authentication, and authorization we can now move on to the last of our four key concepts and definitions, device fingerprinting. Unfortunately, there is no definitive definition in IEEE or RFC for device fingerprinting though RFC 6973 on “Privacy Considerations for Internet Protocols” provides this definition of a fingerprint, “a set of information elements that identifies a device or application instance.” Yet the linking of device and fingerprint remained for “Opinion 9/2014 on the application of Directive 2002/58/EC to device fingerprinting” the EU’s Article 29 Data Protection Working Party, which defines device fingerprinting as “meaning that it includes a set of information that can be used to single out, link, or infer a user, user agent or device over time [93].” Note the opinion which rules, “When a fingerprint is generated through the storage of or access to information stored in the user’s terminal device, the ePrivacy Directive applies [93],” does contain two exceptions. Later in this thesis, I want to return to the exceptions explaining how what I propose does not violate the ruling. We want to focus upon how device fingerprinting is accomplished.

1.3 **The problem of authentication in a dynamic environment**

In the current networking environment, there are many technical issues. Among the many issues facing network managers is the unauthorized device, or the intrusive device within the network structure. Such devices not only provide attack vectors but also consume network resources, creating traffic, introducing latency, and because they are often not managed by a user become a gaping security hole when their OS is outdated, or their software has not been patched to cover known security issues. The second issue a network manager faces in today's poly device network is the authentication of users to the network along with an expectation from the user the corporate network will function much like their home network, with a single sign on (SSO) for all network resources including external connections to additional resources not resident on the corporate network. Many computers and devices still use the username and password combination even though it has been shown to be one of the least effective ways of securing a network. Primarily because both the username and password are subject to so many different kinds of attacks, from shoulder surfing to brute force attacks. Identity theft is not only a risk in terms of personal identity but in the digital world such theft can expose the user to the unpleasant experience of data loss, financial loss, and personal identity loss. Much effort has been expended in creating a variety of solutions to the authentication process including differing two factor authentication, three factor authentication, and four factor authentication.

In two factor authentication, most schemes rely upon self-identification of who you are and often something you know or something you have. In three factor authentication, the scheme may rely upon who you are, something you know, and something you have either a biometric pattern i.e. finger print or a card. Recently, fourth factor authentication schemes

have emerged in an attempt to provide an additional vector of assurance that the self-identified user is who they claim to be. The fourth factor has been identified in several solutions and implementations as location, where you are [1], [17], [77] and [78]. In other implementations, it might be an addition of what you have or what you are.

While each of these schemata have something to offer, they do fall short in terms of full authentication. I want to provide a fuller definition of authentication. As we have seen above the definition for authentication is straight forward and well defined, “The process of verifying an identity claimed by or for a system entity.” I want to further expand that definition by changing the last part of the phrase. Authentication becomes, the process of verifying an identity presented and the device it was used to present. This paper will attempt to suggest a methodology of accomplishing both goals simultaneously.

CHAPTER 2

DEVICE FINGERPRINTING RESEARCH

In a survey of the research literature of device fingerprinting, you can break subject matter down into a taxonomy of four broad categories. The first category is: can we use hardware characteristics to identify a device? The second category is: can we use hardware and software characteristics to identify a device? The third category is: can we use user behavior to identify a particular device? The last category is: can we use the software characteristics to identify a specific device? Further each of these contains the same three subcategories of techniques noted in Kohno's paper: "passive, active, and semi-passive." The key questions all researchers attempt to answer is will the methodology be sufficiently unique to identify the device and if used is it sufficiently reliable to ensure that using the stratagem will not yield significant false positives. In some cases to answer the question the researchers will propose a different methodology for isolating the signature of a device quickly.

2.1 Hardware used to identify a device

In 2005, Tadayoshi Kohno, Andre Broido, and K.C. Claffy published a seminal[1] paper in IEEE Transactions on Dependable and Secure Computing, titled, "Remote physical device fingerprinting." In this paper, instead of seeking to eliminate clock skews, Kohno et al, describes a way to not only exploit the clock skew but to capitalize on it so that a device can be remotely identified. "Clock skews are the inherent tiny drifts in the clocks of hardware devices due to variations in the manufacturing process [8]." To further define this concept, let S_x represent the device's clock and $S_x(t)$ as the time reported by S_x as the true

[1] In "A passive approach to wireless device fingerprinting," Gao refers to Kohno's work as seminal.

time *t*. This is true where $S_x' \equiv dS_x(t)/dt$ and $S_x^{''}(t) \equiv d^2S_x(t)/dt^2$. Let S_y be the sender and S_r be the receiver. Therefore, we can use the terminology as follows:

1. Offset: The difference reported between the time of S_y and S_r, or $S_y(t) - S_r(t)$.
2. Frequency: The rate of clock progression, frequency of time *t* of S_y is reported as $S_y'(t)$. True time is stated as 1.
3. Skew: A skew is the difference between the clock of the sender and the clock of the observer (receiver). So, the skew is relative to the clock of the receiver and is stated as $S_y'(t) - S_r'(t)$.

One final item needs to be defined in clock skew identification, this is drift. Drift is the relative change or shift in a clock skew due to a series of environmental events. This means both the sender's and receiver's clock will drift. In using the clock skew for fingerprinting, the drift must be considered. So, drift is stated as $S_y'(t) - S_r'(t)$. Work has been done to document the relative stability of the drift. Such work has centered on the temperature of the processor and therefore the crystal which is used in the clock.

The clock skew methodology is based upon the following, observations are collected on the timestamp or the clock of the sender. These observations are used to calculate the skew between the sender and the receiver using one of several algorithms, linear programming, linear regression, quick piecewise minimum, or as one paper used "a nonparametric Bayesian method to detect the number of devices and further classify the devices." DF to enhance [62].

One of the major claims in Kohno's work is, "the notion of remote physical device fingerprinting, or remotely fingerprinting a physical device, as opposed to an operating system or class of devices, without the fingerprinted device's known cooperation [52]."

This has all sorts of implications when we discuss later in the paper probable application of device fingerprinting. In the experiments, they tested it against NATs and discovered it worked "even if those hosts use random or constant IP IDs to counter Bellovin's attack, even if all the hosts run the same operating system, and even if not all of the hosts are up at the same time [52]." There are some issues with using the clock skew as suggested by Kohno, in particular the research used the ICMP and TCP packet timestamps. Some of the drawbacks are noted in Kohno, including some operating systems did not use the timestamp consistently. Another drawback noted by a later contributor to the field [68] is the blocking of the ICMP and TCP packets with timestamps with the firewall or the use of NATs which if they do not block the timestamps altogether change the ICMP/TCP timestamp to the same value. (MS site on NAT). While Kohno's work was primarily on the wired side, the wireless side using clock skews was covered by several researchers [8], [34], [46], [54], [68], [74], and [82].

Kohno covered clock skews using the TCP timestamp which per RFC 1323 is a virtual clock that is derived from the system clock. Jana and Kasera covered the use of clock skews for devices on a wireless network in their paper, "On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews," published in 2010. However, unlike Kohno et al which uses the clock from the timestamp Jana and Kasera "use the Time Synchronization Function (TSF) time stamps in the IEEE 802.11 beacon/probe response messages sent by the AP, to determine its clock skew [46]." According to their research this has three advantages. First, beacons are at a fast rate and all the time as well as independent of an application." Second, in RFC 1323, the default difference in clock speed is set in a range of 1 ms to 1 second per tick while the granularity of 802.11 TSF timer is 1

microsecond. This is higher than that of the TCP time stamp clock. "Third, as the beacon time stamp is the actual time when an AP sends a frame (i.e., the time after the channel is sensed to be free) rather than the time when it is scheduled to send the frame, we do not need to consider any significant unpredictable delays incurred by the network as in the case of TCP time stamps [46]." A third method using hardware does not rely only clock skews but takes advantage of the heterogeneity of devices. As explained in, "A Passive Technique for Fingerprinting Wireless Devices with Wired-side Observations," authored by A. Selcuk Uluagac, Sakthi V. Radhakrishnan, Cherita Corbett, Antony Baca, and Raheem Beyah in 2013, heterogeneity of devices uses the uniqueness in device configurations (e.g., processor, DMA controller, memory) and the device's clock skew. This method works well for devices attached to the network that traditionally resist identification or management as all devices attached have some form of the of the trio: processor, DMA controller, and memory. In this method there are four major stages: feature extraction, signature generation, similarity measure, and enrollment. In the first stage, traffic is collected and a feature extraction process is run. The feature extraction uses the packet inter-arrival-time (IAT). "IAT measures the delay (Δt) between successive packets and characterizes the traffic rate. The IAT feature vector is defined as:

$$f = (\Delta\, t_1, \Delta\, t_2, \Delta\, t_3, \dots, \Delta t_i)$$

Where Δt_i is the inter-arrival time between packet i and $i - 1$ [82]." The second stage is signature generation. In this stage, statistical analysis is to reveal patterns in the measurements. Uluagac et al used a time-domain model measuring the distribution of the IAT feature. The distributions capture the frequency event of stage one and tosses them into time defined bins, B_n. These bins are equally spaced across the time defined space. "The

device signature is sensitive to the bin width and different bin widths will reveal different information about the feature vector [82]." If the bin widths are too small, then fewer IAT values will occur within the specific bin. This causes true features to be falsely identified as random noise within the traffic. However, if the bin is too large, information that is critical might go missing causing a device to be falsely identified. Empirically the authors determined the best bin size was where the time value for the bins was 300, $n = 300$. Once a signature is generated the third stage, similarity measure, to check for similarities against a master database using a neural network to determine a match value ranging from 0 to 1 where 1 is a perfect match. The results are passed on to stage four, enrollment. In this stage the signatures are passed through two neural nets, one for device type and the second for device id. In this case, there are three possible outcomes: recognized device and type, recognize the type but not the device, do not recognize the type or the device. The biggest drawback to this methodology appeared to be the scalability of the model which experienced a linear decrease in the successful identification of a device in correlation with the number of devices on a network as shown below in the graph reproduced from the paper.

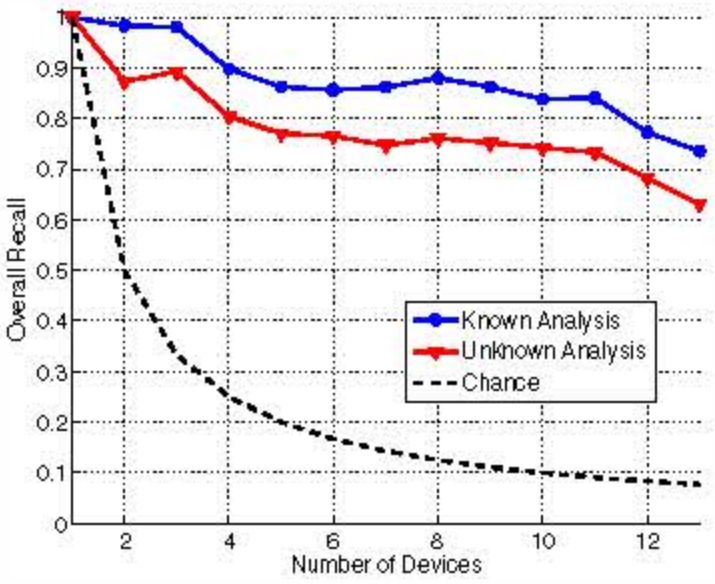

Figure 2 – Recall vs Number of Devices

As the authors note, "The two most important factors that affect time associated with GTID's decision process are capture time and processing time [82]." Increasing the sample size resulted in increased time for capture. And if the packet size or sample size is increased then the processing time of the sample size also increases. Efforts made to reduce the time by multi-threading the methodology did result in a decrease in the over-all time but still reflected an increase in time from the smaller packet samples.

Another methodology for device fingerprinting in a poly network is using sensor fingerprinting to identify a mobile device. Bojinov, Boneh, Michalevsky, and Nakibly cover this approach in their paper, "Mobile Device Identification via Sensor Fingerprinting."

Most mobile devices i.e. tablets and smartphones today include the following features, speakers, microphone, accelerometer and can be divided into essentially two major eco-systems Android or Apple (iOS). In Bojinov, et al, noted there are multiple features within the mobile device area that have slight imperfections that can when aggregated allow

for device identification. These features and their type of imperfections, a linear bias, tolerance, or timing are noted in the table below.

Table 1 - Mobile Device Sensors

Sensor	Imperfection
Audio	Tolerance gain
Accelerometer	Linear bias
Gyroscope	Linear bias
Magnetometer	Linear bias
Ambient light	Linear bias
GPS	Clock skew
Touch screen	Misalignment
Camera	Pattern noise

Linear bias is defined as true value versus measured value. Where the sensitivity *S* of the device being measured and the offset *O* of the sensor form part of the measurement. In theory, Sensitivity should be equal to 1 and offset should be equal to 0. However, in practice there are small imperfections that are measurable so, measure value equals true value with sensitivity and offset as factors, thus you end up with $v_m = v_t S + O$. After reviewing the various sensors, the research team determined the results of several sensors are not immediately available through the mobile connection, however microphone-speaker and the accelerometer results could be accessed through the wireless connection.

When using the microphone-combination, “the fingerprinting system uses the speakers to emit a sequence of sounds at different frequencies and records the resulting signals using the microphone. The fingerprint is computed by looking at amplitude and frequency distortions in the recorded signals [11]. Since the observer is eliciting a response form the device, this is an active method of fingerprinting. The central measurement of microphone and speakers is the frequency response. In both, the frequency is normalized over a given range, in the case of the microphone the normalization occurs in

the output gain, in the speaker equally output audio intensity is normalized. Again, due to slight manufacturing imperfections the frequency response of microphone-speaker is not identical. Manufacturing tolerances are such that a typical tolerance is ±2db and it is this various that allows for the use of the microphone-speaker combination. This method is a little unique for device fingerprinting and does not work unless the device is equipped with both a microphone and speaker. Because you will stimulate the speaker to produce a sound and record it with the microphone and note the difference between the original intensity and the recorded intensity, this is called the feedback ratio. The whole process looks something like the figure below:

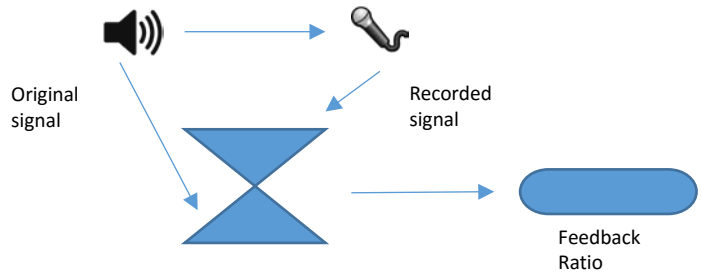

Figure 3 - Measuring Feedback Ratio

Using this method, a specific harmonic frequency was noted to produce the correct tolerance so as a specific machine could be isolated. This frequency was recorded in three different acoustical environments allowed the team to record signatures for the device. These signatures were then enrolled in a database. "Given enough samples, the group was able to estimate the mean and variance of the distribution [11]. When the device was encountered later the team could correctly identify it 95% of the time. Of course, the main drawback to this method is the use of frequencies the device is stimulated to produce. It would be hard to

stealth this method unless using a set of frequencies that were either above or below the normal hearing range. The accelerometer method does not suffer from this drawback.

As you are aware, the accelerometer measures force along a three-axis continuum allowing the device to adjust the screen to either portrait or landscape as well as serve as a pedometer. An advantage of using the accelerometer is in normal use the user will place the device on a stable surface creating a given value, *g*. This is observed because the accelerometer value is relatively static. "What is more, iOS as well as Android browsers expose this functionality to websites without notifying the user [11]." Unlike the microphone-speaker combination, there is no way to feed values into the accelerometer. Instead the passive observer must rely on the user moving about and collect enough observation to estimate the accelerometer's calibrations. The first measurement to be collected is the effect of gravity on the device since "the *Z* axis will register practically all the acceleration due to Earth's gravity [11]." Since the device is unknown two measurements are needed, device face up Z_{m+} and device facedown Z_{m-} this will allow you determine the value for S_z, or the sensitivity along the *Z* axis. We can do this by using the following:

$$S_z = \frac{Z_{m+} - Z_{m-}}{2g}$$

$$O_z = \frac{(Z_{m+} + Z_{m-})}{2}$$

Then to find the accelerometer bias, deviation from true value, multiple measurements were made, $X_m^{(i)}, Y_m^{(i)}, Z_m^{(i)}$ as many as could be gathered and then fed through the optimization problem

Minimize $\sum_i \varepsilon_i^2$ subject to the following as reproduced from Appendix B of the research,

$$((X_m^{(i)} - O_x)/S_x)^2 + ((Y_m^{(i)} - O_y)/S_y)^2 + ((Z_m^{(i)} - O_z)/S_z)^2 - g^2 = \varepsilon_i$$

Where *i* is the index measurement. Applying this to multiple sets of measurements for each device allows for labelled samples in 6-dimensional space. “Then use nearest neighbor matching (KNN) to associate the sample with a labeled cluster. Cross-validation of KNN classification over this data yielded a correct classification percentage of 81.3% [11].” As you can note there is a high probability of device identification using these two methods for mobile devices containing at a minimum a microphone/speaker and an accelerometer with the drawback that one will emit frequencies and the second will require extended periods of observation. In the end, the hardware based only method has some drawbacks as was noted in covering the research into this methodology.

2.2 Hardware and software characteristics to identify a device

Another method of device fingerprinting includes the use of a hybrid of hardware and software to identify a device. In “Active Behavioral Fingerprinting of Wireless Devices,” the team of Bratus, Corneliius, Kotz, and Peebles (Bratus, et al) propose a solution based upon the chipsets of wireless cards, their drivers or the firmware. In their solution, they send malformed 802.11 frames and observe the responses, claiming the responses different sufficiently to allow for device identification. This approach is based upon the concept that different implementations of the 802.11 standard will respond differently and their research appears to back up their claim. This is also an example of the active technique of device fingerprinting in that the observer causes the device to send information which can be used in its identification. This same method of using the device coupled with driver behavior is also advocated in “Passive Data Link layer 802.11 Wireless Device Driver Fingerprinting,” authored by Franklin, McCoy, Tabriz, Neagoe, Van Randwyk, and Sicker (Franklin et al).

Their approach is completely passive with the only requirement being the attacker "needs to be able to monitor the wireless traffic from the fingerprintee [27]." It makes use of lack of an explicitly defined algorithm for scanning for access points (AP). "This lack of an explicit specification for a probing algorithm in the 802.11 standard has led to the development of many wireless device drivers that perform this function entirely differently than other wireless device drivers [27]." The approach is also unique in that it does not require a device to be connected, it just requires the device to be probing for a connection. The approach uses a two-stage process. During the first phase, the observer monitors for 802.11 traffic, capturing it in a trace. In the second phase using a supervised Bayesian approach the information is first binned. This allows for smooth of the data. The team then chose two values of specific interest during the data analysis phase, "frequency of delta arrival time values between probe request frames. The second attribute was the average, for each bin, of all actual (non-rounded) delta arrival time values of the probe request frames placed in that bin [27]." This allowed for attribute characterizing the size of the bin and a second attribute describing the mean of the bin. The signature for each driver was created by recording the percentage of all probe request frames in each bin combining it with the average of all actual delta arrival time for the probe request frames in the bin. This value is then placed into the master database of signatures. Once this is done then you would calculate closeness to see if the trace you have captures has already been identified.

To calculate closeness on the trace you have acquired, first create a signature, S using the method described above. Then you will let p_n be the percentage of the nth bin of S. The mean value of all probe requests in the nth bin is expressed as m_n. From here we will use D to represent all the signatures in the master signature database, with d standing for a single

signature from the database. Let y_n and x_n respectively be the percentage of probe request frames and the mean of all probe request frames in the *n*th bin of *d*. Using the following equation you will calculate closeness, *C* representing the distance value.

$$C = \min(\forall b \in D \sum_{0}^{n} (|p_n - y_n| + y_n|m_n - x_n|))$$

This is run reiteratively through all the bins in *S* summing the difference of the percentages and mean differences scaled by the percentage. The mean differences are scaled by the *d* bin percentage to prevent this value from dominating the bin percentage differences. This provides success in appropriately identifying the signature of the wireless driver in the trace ranging from 96% accuracy to 77% accuracy. The highest percent of accuracy obtained with a large data set and relatively uncluttered environment with the lowest result obtained through a very noisy environment cluttered with multiple signal sources.

Both of these methods rely upon the 802.11 frameset which may limit the effectiveness to wireless devices and while the wireless sea of devices is constantly expanding when addressing network security any solution would need to take into account what I have called a poly network.

2.3 User behavior to identify a device

The third method while not primarily used for device fingerprinting can be used to associate a user with a specific device. This methodology relies upon user behavior and is predicated upon the concept that user behavior can be predictive and unique. "Adaptive Authentication based on Analysis of user Behavior" authored by Abu Bakar and Haron illustrates the use of this method for the purposes of authentication. A second interesting paper in this area of research is titled, "802.11 User Fingerprinting." In this paper the authors

show there are four implicit identifiers that allow for identification of users even with MAC addresses hidden and user ID and IP addresses not available. These four identifiers are: network destinations, network names advertised in 802.11 probes, differing configurations of 802.11 options, and size of packet broadcasts. Using these identifiers, the researchers were able to develop a tool to automate the process of identification and with a high degree of accuracy identify individuals during a conference. Furthermore, this tool worked even when the link layer was encrypted. With this methodology, several hundred users were explicitly identified while attending a major conference. In reviewing the sample of information collected *s* during the conference the team asked a simple question did this traffic come from a specific user? To answer the question, the team of Pang, Greenstein, Gummadi, Seshan, and Wetherall established a classifier for each user, C_u. If the traffic sample *t* comes from the user, then C_u returns a yes, else no. So, from each traffic sample a vector of features was extracted, $(f_1, \dots, f_m)$. "For each feature f_i, they estimate the posterior probability distribution Pr[*t* has f_i |*t* is from *U*] and the prior probability distribution Pr[*t* has f_i] from training data."[p6] Using a second formula if the value is calculated to be higher than the threshold *T* then the sample is from *U*. This formula is displayed below, remember the key question is did the traffic sample *t* from a specific user and does it have $(f\_1, \dots, f\_m)$. So we end up with Pr[*t* is from *U*|*t* has$(f_1, \dots, f_m) =$

$$\frac{\prod_i^m (\Pr[t\ has\ f_i | t\ from\ U]) \cdot \Pr[t\ is\ from\ U]}{\prod_i^m \Pr[t\ has\ f_i]}$$

Is this value is greater than the threshold *T* then this sample is from *U*. From there using the training data, a user profile is constructed $Profile_u$. This profile contains all the elements in union with the training sample from the user *U*. In addition, some features have more weight

than others based upon the occurrence, reoccurrence within a user's traffic sample. These values are then compared with the feature found in the user profile.

The drawback to the method is the threshold. Without a priori knowledge, the adversary has to select a threshold that results in a false positive result that is acceptable to the attacker. This work will re-manifest itself the related topic, authentication or re-authentication and in chapter three of this paper we will see how user behavior can play a role in continuous re-authentication of the user using a device already fingerprinted. As a side note, user behavior is a little explored are of fingerprinting for purposes of device identification. User behavior has been explored in authentication schemes and we will review this some of this literature in our quest to provide a secure, usable solution that accomplishes the definition for authentication provided earlier the process of verifying an identity presented and the device it was used to present.

2.4 Browser or software used to identify a device

The fourth methodology for device fingerprinting involves chartering the software differences. This can be done effectively by using the common interface between the internet and the device, a browser. In their paper, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting," Nikiforakis, Kapravelos, Joosen, Kruegel, Piessens, and Vigna (Nikiforakis et al) explore the world of browser fingerprinting, how it is used to identify a device and by extension the user, and how pervasive and persist across the Internet. From 1994 until now cookies have been a part of the browser eco-system. They are prevalent today as you can see from clip of a screen shot taken using Windows 10 Edge browser of a popular news site.

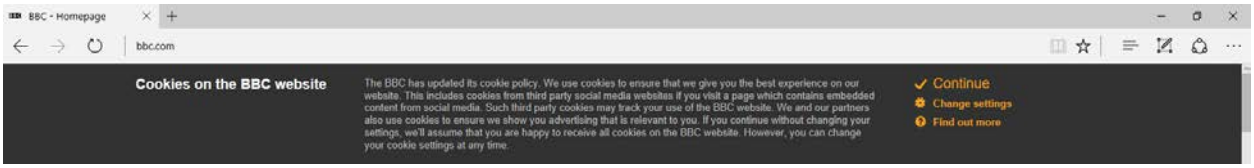

Figure 4 - BBC's statement of their cookie policy

Spread throughout the Internet sites, the use of these files to track users across multiple domains, and sites and "their direct connection with online behavioral advertising, captured the attention of both the research community and the popular media outlets and, ever since, cause the public's discomfort [63]." An EU workgroup set up under the aegis of the Directorate of Fundamental Rights and Union Citizenship, ruled in 2002 that cookies violated a citizen's fundamental right to privacy. And in 2009 went so far as to say, that before a site can use cookies they must obtain a citizen's prior consent. Browsers even incorporated a setting for deleting cookies upon exit from a browser. These developments pushed advertisers and data aggregators to find new and different ways to track users across the web and from website to website. When it was uncovered, that browsers are fairly unique from device to device the concept of exploiting this uniqueness was advanced. Today, there are several commercial firms, Qualia, Iovation, and Threatmetrix that boast of large databases of device fingerprints using the browser eco-system as their methodology for developing each unique fingerprint. The three primary methods of using the browser echo systems exploit features available not only in the browsers but also in popular add-ons such as Adobe Flash and Java. Peter Eckersley, in his paper "How Unique is your Browser?", presented in 2010 at the International Symposium on Privacy Enhancing Technologies Symposium estimated that out of a sample size of 470,161 the odds of two having the same signature had odds of 1 in 286,777. The odds increased if Java or Adobe Flash was present.

The table below reflects the items that were measured to determine browser uniqueness and thereby device uniqueness.

Table 2 - Browser Measurements used in Panopticlick Fingerprints

Variable	Source		
User Agent	Transmitted by HTP, logged by server		Contains version, info.
HTTP ACCEPT headers	Transmitted by HTP, logged by server		
Cookies enabled	Inferred in HTTP, logged by server		
Screen resolution	Javascript		
Timezone	Javascript		
Browser plugins, plugin versions, and MIME type	Javascript		Sorted b
System Fonts	Flash/Java applet		Not sort
Partial Super cookie test	Javasxcript		No test f cookies. DOM G

Later research by Nikiforakis et al revealed additionally, “the order of property-enumeration of special browser objects, like the `navigator` and `screen` objects, is consistently different between browser families, versions of each browser, and, in some cases, among deployments of the same version on different operating systems [64].” This adds additional information that the observer can use to fingerprint a specific browser on a specific device and by extension the device itself.

As we can see from this chapter, it is possible to uniquely identify the device whether through clock skews, or a combination of hardware/software, user behavior, or browsers. In some cases, the behavior of the user is such that we can identify the user through passive data collection. In today’s modern networks with their multiple devices and with some users having several devices that they use for their connectivity, devices ranging from smart phones and tablets to the traditional workstation or laptop, it is important to be able to

identify the not only the device but if possible the user. Not only is it important to identify the device but to tie the device to the user so that each user can access the resources they need regardless of the device they use to access the network.

CHAPTER 3

APPROACHES TO AUTHENTICATION

The process of authenticating an identity has been the point of attack of attack since earliest times. In the movies, the sentry calls out, “Who is there?” The user replies with his name at which point in the protocol he is challenged to provide the sign for that night. This was so only authorized soldiers, and not the enemy were allowed into the circle of trust, the camp. In todays’ digital world, we often approach a workstation to login into our work and the computer provides a screen that in effect says, “Who’s there?” We enter our credentials, whether it is a user name or a token and immediately are challenged to produce the password, our sign. Some sites, require the user to change their password every 30-60-90 days depending on the anxiety of the security personnel. Others require passwords that are not words and must have a mix of numbers, special characters, uppercase and lowercase letters. Still other sites are fine with a password that is a minimum of 8 characters long. Access to the internet has grown and exposure of services to the internet has exploded over recent years. “An average user accesses about 20 different accounts in a day, which are generally accessed through password based authentication mechanisms. It has become overwhelming for users to remember so many passwords and therefore they have resorted to typically three main approaches to overcome the friction: 1) users use the same password across multiple SPs; 2) users note their passwords on a piece of paper or their personal mobile devices; and/or 3) users use shortcut helper functions such as “remember my password” features on browsers [73].” As the apparent requirement for a stronger authentication mechanism has grown, various solutions have been offered. In this chapter I will briefly review the terminology of multi-factor authentication, define the process, review the research, the

extensible authentication protocol, and review one of the remaining issues in most authentications schemes.

3.1 Terminology for multi-factor authentication

In discussing multi-factor authentication there is common terminology. But, as you read the literature covering this area, it quickly becomes apparent that not everyone means the same thing when using certain words. Words such as authentication can be narrow as in "Authentication Systems of Internet of Things," where it is defined as "Authentication is the process of confirming one's identity;" or expanded as in "A User Authentication Protocol based on Multiple Factors," where "User authentication is a process of verifying the user's identity with reliable methods." Then while there is general agreement in the existence of such terms as two-factor, three factor, and four factor authentication; where a particular methodology belongs can be a source of disagreement. One set of researchers classify username/password as a single factor authentication, the username is the identity to be verified. Others place the same methodology in two factor authentication insisting the username functions not only as an identity but also as a factor of authentication. Clarity as to which it is, single or two factor is obtained by understanding what is meant by the four different phrases.

The best statement regarding this begins with understanding the four factors, "Authentication is classically divided into four distinct categories: *Something You Are*, e.g., biometric authentication; *Something You Do*, e.g., behavior based authentication; *Something You Know*, which relies on information from user memory; and *Something You Possess*, which typically extends protection by verifying possession of some physical element such as

a key fob, for example [73]" For the purposes of this paper I want to define the four factors providing a consistent frame of reference. For the purposes of multi-factor authentication, a single factor authentication is simply presentation of a credential to validate a previously produced identity. In two factor, multi-factor authentication, it is the presentation of two separate credentials. In three and four factor, multi-factor authentication, it is the presentation of three and four, respectively, sets of credentials. Note nothing is said that the credentials must be separate and unique in the classical taxonomy. Thus, it is possible to have two sources of what you know, of three sources of what you know, or a mixture of what you know, who you are, and what you have.

As we have defined authentication earlier using the IEEE definition, we need to understand the goal of authentication. In NIST SP 800-63-2, Electronic Authentication Guidelines, there is a clear statement of the purpose of authentication. "Authentication establishes confidence in the Claimant's identity, and in some cases in the Claimant's personal attributes (for example the Subscriber is a US Citizen, is a student at a particular university, or is assigned a particular number or code by an agency or organization). Authentication does not determine the Claimant's authorizations or access privileges; this is a separate decision [14]." Claimant in this case is the entity, whether person or machine presenting the identity. In the next section, we will explore the process used to accomplish the goal.

3.2 Authentication as a process

In the introduction, we discussed how identification, authentication, and authorization and a map to illustrate the process. We are now going to look at the sub-process, identified with a darker shade of color in the figure below:

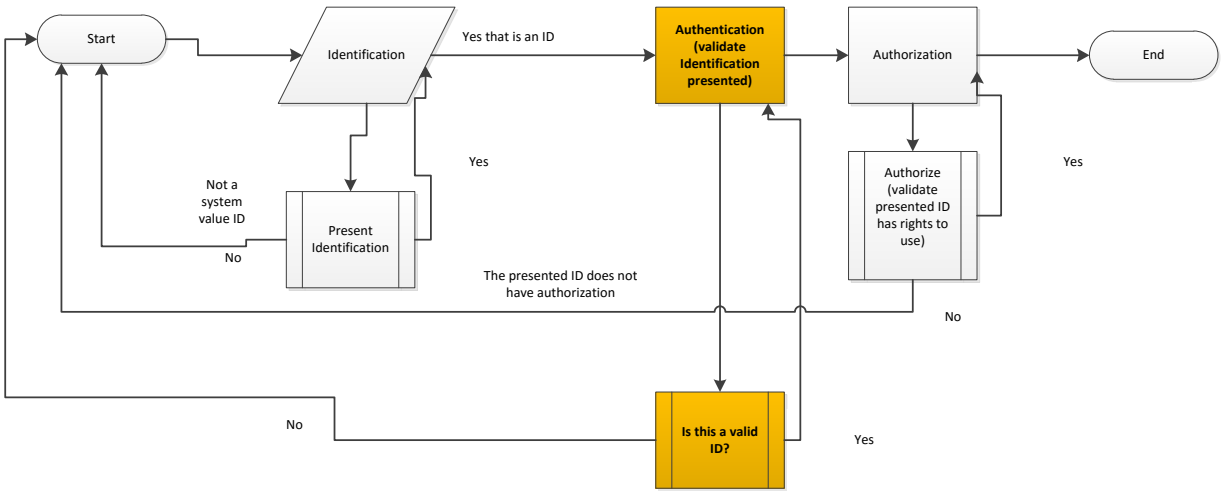

Figure 5 - Authentication sub-process

This part of the process consists of several actors; we have the requestor or claimant. This is the entity presenting the identification for validation. We have the entity seeking verification of the credentials presented, in NIST SP 800-63-2 referred to as the verifier. The information is referred to a third party for verification the claimed identity is valid and belongs to a subscriber, a legitimate user of the information system. This third party maybe called a relying party (RP) or a credential service provider (CSP). The relying party will review the information provided, valid that information against an internal source, often this is a database of credential information, and the provides a yes/no validation response to the verifier. If the answer is no, a token for authorization will be granted and the claimant validated as a subscriber is able to use what resources they are authorized. In some cases, the RA is not the CSP in those cases the RP acts as a middle entity passing the information both ways isolating the verifier from the CSP. The whole process as described above is seen in this next figure which comes from the NIST publication.

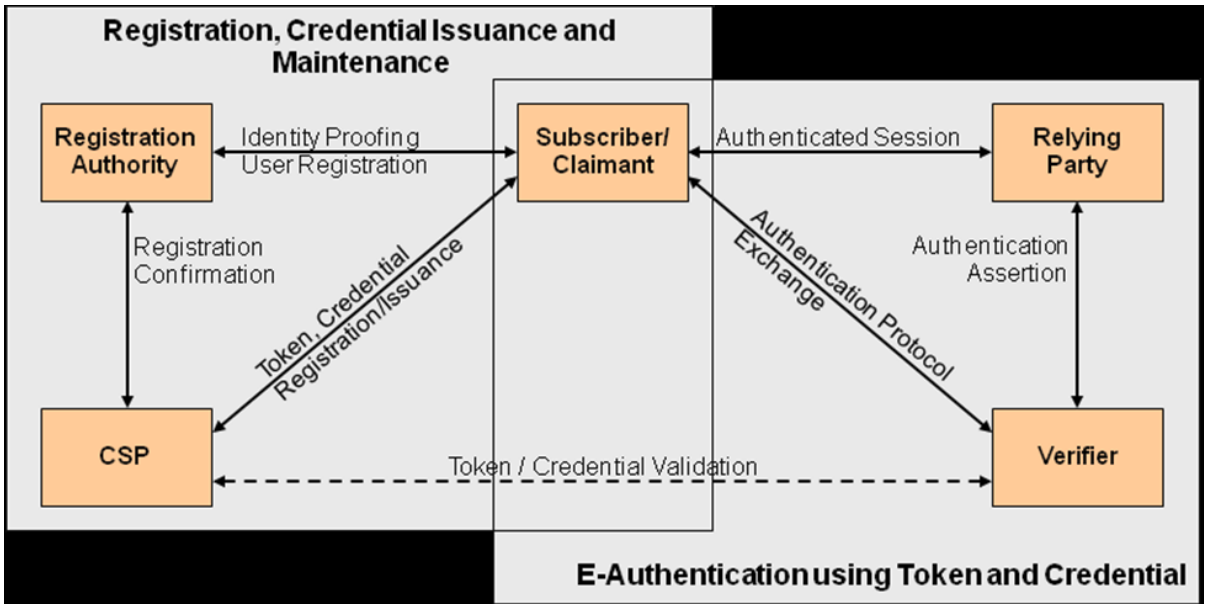

Figure 6 - NIST Authentication Process

The CSP is required to maintain information about the Claimant. In some cases, the information has an expiration date on it. Other cases, this information, while not considered a good security practice, have an unlimited life. Applying multi-factor authentication to this process you can see that the validation process in effect validates each piece of information collectively. In some cases, there may be multiple CSPs but in most cases the CSP is a single entity.

3.3 Research

There is a wide variety of research occurring in multi-factor authentication. Some of that research occurs in each of the areas of multi-factor authentication: knowledge-based, possession based (biometric or token), behavior based, and physical based. This paper will explore some of the relevant information currently being advanced. Any good authentication process will have to take into account a major stakeholder, the users. A recent internal study by Intel revealed, “users desire security but not at the expense of convenience

[77]." In, “A Multi-factor Re-authentication Framework with user Privacy,” it was noted there are essentially three user requirements for any authentication method. The first is, the method should be user-friendly i.e. not forcing the user to re-enter primary factor, or a password, at intervals. The second requirement is it should protect the user’s privacy even if the method is based upon user characteristics and those characteristics are stored on authentication servers. The last factor is the method should be fairly quick in its resolution of authentication. Simply put any solution *S* will have to ensure the user’s discomfort level *U* remains below a threshold of 1 where 1 is the level at which the user will resist implementation or implement their own work around, $S = 0 < U < 1$. With that we begin looking at some of the research in authentication.

Following our types of factors provided earlier we will look at, knowledge based as a factor in the authentication. In their paper, “New Factor of Authentication: Something You Process,” the team of Shakir Ullah Shah, Fazl-e-Hadi, and Abid Ali Minhas suggest a form of the puzzle solving for user authentication. While they call this something you process, for easy classification purposes we will place it in the knowledge based authentication. This is called formula based authentication. In this method, the user self identifies and the system then provides a formula which varies upon each log in. For example, the system would present the formula, A+B-C = 14. The user resolves the formula, providing their answer as part of their presentation of credentials. One of the pros the authors present for their system is that “It saves the time wasted by entering wrong password as humans do by allowing the user to see the result of the formula [72].” I would argue that those who are math challenged or numerically dyslexic would find this type of methodology challenging.

In the area of possession based authentication, there are several papers. In developing their concept, the team of Chun-I Fan and Yi-Hui Lin, in their paper titled, "Provably Secure Remote Truly Three-Factor Authentication Scheme with Privacy Protection on Biometrics," are responding to two problems with biometric characteristics used in remote authentication. The first is, some biometric characteristics may be easily obtained and they can never be changed, which makes these biometric characteristics unreliable as encryption keys. Second, the biometric capture devices are remotely located. The server cannot check whether the device is capable of verifying that a person is alive [24]." In their solution they propose a three factor authentication method that contains, something have, something you know, and something you are. In our authentication taxonomy, this is reduced three factors but physical (something you possess), knowledge based, (something you know), and physical (something you are). The method has three phases, 1) initialization, 2) registration, and 3) login and authentication. In the initialization phase the user is not an actor, the server prepares a set of security parameters. Including in this is a private-public key set (*sk*,*pk*) and a secret key set (*x*,*sk*). This is required in order to keep the user biometric data secure. In phase two the user is a very active agent. In this phase the user will execute five steps registering to the server. In step one of the registration the user U_i chooses a random string *r* and picks a password PW_i. Scans their iris and creates a template, S_i. So $U_i\ computes\ SS_i = \delta_r(S_i) = r \oplus S_i$. In step two sends the password and the biometric template to the server using a one-way hash. In step three the server will compute and ID and store in the ID table. At this point the server will send the user a smart card. In step five the user using the smart card will encrypt the random number from step 1 with the biometric template and stores the resulting sketch in the smart card. In the login and

authentication stage the seven steps. In step one, the user will login using the password and allow for an iris scan S_i^*. Then using the smart card retrieves *r* from the sketch and then computes the iris scan and using the ID and the public key from the server encrypts deriving C_0. In step two the user sends the derived value to the server. The server decrypts the derived value using its secret key. Obtains the ID and the iris scan, compare the values to the values stored in its ID table in step three. In the fourth step, the server then computes a new derived value and sends it to the user. The user in step five will use this value to extract a value checking to ensure the value came from the server and finally gets a value, *v*, to be used in a session key. Then the user computes a new derived value, C_2. In step six the user sends the new derived value to the server. Finally, in step seven the server will use the value computed in step three from the derived value, to decrypt the newly arrived derived value, making certain this was sent by the user. Then it checks to see if the hashed password sent matches the hashed password stored in the registration phase and if the score between the two iris scans meet the predefined threshold value set. If so the server returns true accepting the login in request. Using this methodology, the team contents it has accomplished four goals, message content authentication, message origin authentication, general identity authentication, and session key material provided.

A paper in the physical based area is titled, “Four-Factor based Privacy Preserving Biometric Authentication and Authorization Scheme for Enhancing Grid Security” from the team of G. Jaspher Willsie Kathrine and E. Kirubakaran builds on the work above by adding a fourth factor, someplace where the user is, also known as geographically based authentication. This type of approach is to “enhance the security criteria required for a vast distributed system” i.e. the military, banking, medical, or research environments. The paper

refers to such organizations as virtual organizations (VO). It also adds a security framework for this grid of virtual organizations. The security component it proposes to add is three main subgroups. A security client (SC), a security manager, (SM), and a chief security manager (CSM). In adding the location feature to the previous framework, the team proposes adding to more variables to document these features, the first is the dynamic user ID (*CID*). The second variable is the dynamic service ID, (*SID*). Then adding two final variables to handle the geographic location, (pos_i) for the location information attached to the user and (pos_s) for the location information attached to the server. The three phases remain essentially the same as the previous paper except the login and authentication phase are separated. This because the new variables are added in this last phase. So in the first step of the last phase where the user created a derived value from the following formula: $C_0 = e_{pk}(ID_i \| y_i \| u$ where u was randomly chosen by the user and e_{pk} denotes the server's public key encryption function using the public key from the server the paper adds the nonce pos_i so the formula for the derived value now looks like this: $C_0 = e_{pk}(M \| ID_i \| y_i \| u \| pos_i)$, the smart card previously computing M as $M = (K \oplus n_u)$. In step three of the previous paper the server now computes a derived value as follows: $C_1 = E_u(N \| SED \| S_{ID} \| v \| pos_s)$. The added values to the formula are the computer generated nonce $N = (K \oplus n_s)$ and the server's position pos_s. This contains each time the derived value is calculated, adding a nonce to prove liveness and the positional data as an additional authentication value. In effect the authors added two new values, not just the geographic one. The paper shows how using the calculated nonce as part of the user ID this protects against the ID theft attack. Additionally, the time performance of the four factor authentication scheme was also reviewed and replacing the one way hash with the XOR function saves time.

There is additional work using location based services as a fourth factor in multifactor authentication. In some they use the GPS position [1] and in others they use a signal property [78].

The next set of work on multi-factor authentication is behavior based. Behavior based encompasses a range of solutions, using the user behavior. Two of our resource papers refer to user behavior by using the mouse and keyboard. A third paper takes a more macro stance in regards to user behavior. One of the more interesting thoughts in regard to user behavior is the use of this approach in a form of continuous authentication, or reauthentication. This type of use would address a perceived weakness in the current model which is; once authenticated, how does the system ensure the entity originally authenticated is the entity that uses the services and resources of the system. In their paper, "Combining Keystroke and Mouse Dynamics for Continuous User Authentication and Identification," Sourmik Mondal and Patrick Bours (Mondal and Bours), introduce two concepts, static authentication (SA) and continuous authentication (CA). As they define SA there is a once time authentication and "the legitimacy of the user is assumed to be the same during the full session." The weakness to this approach one of them is the user leaving the system unlocked while no one is there. The suggested solution is a form of ensuring the entity that authenticated is the same entity for the whole of the session. Mondal and Bours call this CA. They accomplish it by continuously monitoring the entity's (in this case the user's) behavior during the session based upon a biometric signature left on the device. As they point out this is not an alternative to SA but CA is intended to complement SA. As was pointed out earlier users want security but not at the point of being inconvenienced. This method is unobtrusive due to the nature of the behavioral biometrics; keyboard dynamics (KD) and mouse

dynamics (MD). The first step in this is feature extraction. This is achieved by encoding every keystroke *k* as $k = (A, T^p, T^r)$ where T^p,T^r are the timestamps of the key press and release in milliseconds and *A* is the value of the key pressed. These events are then converted into two different actions, one, single key action and the other is key digraph action. In the single key action, the feature vector is $FV_i^{sk} = (A_i d_i)$, where A_i is the i^{th} key, and d_i is the duration of the i^{th} pressed key (i.e. $d_i = T_i^r - T_i^p$). In the diagraph key action, two consecutive keystrokes are considered a diagraph action when the latency between the two is less than 2000*ms*. The diagraph key action feature vector is represented by $FV_i^{di} = t_i, l_i^{rp}, l_i^{rr}, l_i^{pp})$, where t_i is the total time duration of two consecutive keystrokes. l_i^{rp}, l_i^{rr}, and l_i^{pp} are the latencies between the i^{th} and $(i+1)^{th}$ keys such that $l_i^{rp} = T_{i+1}^p - T_i^r, l_i^{rr} = T_{i+1}^r - T_i^r, and\ l_{i+1}^{pp} = T_{i+1}^p - T_i^p$. Each mouse event is converted to four different activities, mouse single click, mouse double click, mouse move, and mouse drop-drag. Mouse single click measures the duration between mouse click and release and is similar to the single action. The double click is similar to the diagraph key action only the duration between the two clicks has a threshold of 1000*ms*. The mouse move is a sequence of mouse move events. The mouse drop and drag while similar to the mouse move event must first begin with a mouse click and followed by a mouse release. The next section to provide training data sets for each user. In their experiment Modal and Bours found that one data set was insufficient to achieve desired results and they required multiple training data sets for each user. Once these sets were provided to the continuous authentication system (CAS) and the continuous identification system (CIS) they were able to achieve a measurable success rate. CAS worked with the notation that it took an average of 471 actions to detect and imposter with an average accuracy of 61.3%. The CIS was able to accomplish its target with fewer events,

333 actions on average before it detected the imposter with an average accuracy of 58.9%. Another finding was addition higher accuracy was achieved using the KD versus the MD.

A similar paper, "A Novel Approach to Design of User Re-Authentication Systems,' by Harini Jagadeesan and Michael S. Hsiao (Jagadeesan and Hsiao) uses the same KD and MD of the previous authors but adds an additional constraint during the training data set, using pre-determined tasks. In addition, while Mondal and Bours collected four events for the mouse Jagadeesan and Hsiao collected 10 events, including mouse speed and acceleration in all directions. After using the broad indicators general speed and acceleration to classify a user, the more specific mouse movement (directional speed) allowed for a narrower identification of the user. The use of the keystroke diagraph was limited to 50 of the more commonly occurring keystroke diagraphs. These keyboard diagraphs were accomplished by choosing ten different articles from popular magazines and a frequency analysis was done. The two sets of events were also compared to produce a mouse to keyboard interaction ratio. The framework of their method consisted of two different engines, the first is an environmental engine which collects the data from the keyboard and the mouse. These values are then passed to the analysis engine. "The analysis engine applies three heuristics to data passed to it and provides the most probable user match from the database [45]." The user match is provided back to the environmental block which then validates that this is the "logged in user". If there is a non-match, then the event is logged. After a certain threshold of non-matches is reached the user is logged out and must begin the authentication process. Just as Mondal and Bours had to do this framework also requires two data sets for each user. The first data set is from the constrained training phase where the user performs a number of pre-defined tasks. After a profile is established during this phase, the second phase,

verification is entered. During this phase the user is no longer constrained but independently accomplishes tasks for a pre-set period of time and the results are collected in a profile. The combination of the profile obtained from the two data sessions, training and verification becomes the user profile. While accuracy of their framework was high it was noted that the increased number of users led to lower accuracy results in some cases. One other conclusion was provided of note, that this type of framework will require periodic re-training to account for the subtle changes that occur in a person's behavior over time.

Another set of researchers, Khairul Azmi Abu Bakar and Galoh Rashidah Haron (Bakar and Haron), wrote a paper, "Adaptive Authentication base on Analysis of user Behavior." This framework uses an authentication system called Unified Authentication Platform (UAP). The UAP architecture is depicted in the figure below:

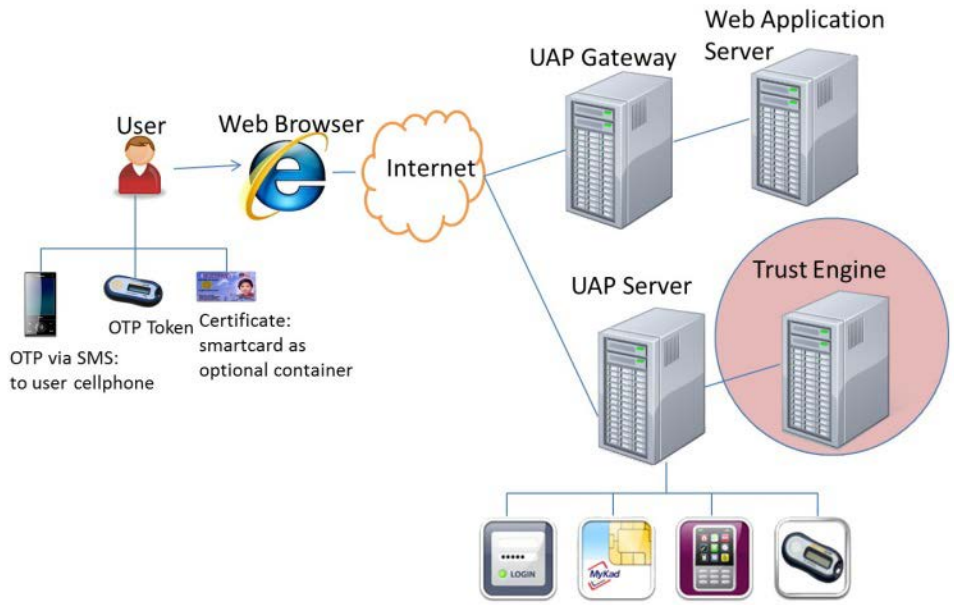

Figure 7 - UAP architecture

Adaptive UAP has two primary processes, pattern generation and trust evaluation. During the pattern generation process the user's behavior from past logins is analyzed and a profile is developed. This is accomplished by reviewing the event storage what has context records

about each user. In this study's case the primary context records used were: the login time, user geolocation, application accesses, the type of browser, and the operating system. Each of these items were given a specific weight. The pattern generation process is a reiterative process that runs primarily during non-peak low usage hours as it consumes a lot of computer power in reviewing all the past records from the last 14 days for all the users in the system. In this manner, you have a constant refreshing user profile based on recent user behavior.

The second process is the trust evaluating process. This process occurs upon every login request form a user. Containing five components, context collector, patterns storage, trust calculator, challenger, and events storage, the trust evaluation will run through the process using the time clock from the server as part of its process. Central to the process, the trust calculator "compares the current contexts processed by the context collector with the user attribute profiles retrieved from the pattern storage to decide the total trust score of the user [3]." From there the challenger component will review the user's score form the trust calculator, matching it against the threshold for a specific application the user wants access to and provide one of three response, grant access, request additional input for user, or block the user.

While the UAP framework is adaptable and flexible it has a high overhead in that as the system gains more users, the number of records stored increases as does the processing time for creating the user profile. This would argue again the framework being scalable to an enterprise level.

3.4 Extensible Authentication Protocol

As part of the survey in the research ongoing in the area of entity authentication, we need to look at the use of the Extensible Authentication Protocol (EAP). The primary reason

for doing this is because EAP is concerned with the three As in network usage, authentication, authorization, and accounting. For the first one alone, authentication, we would need to review this literature. Because the solution we posit for a variety of issues in network authentication and usage will require the use of one of the various methods that spring out the EAP standards. To do this I am going to look at three items, two of them research papers and the last is the RFCS for EAP in particular the following, RFC 3748, RFC 4962, RFC 5216, and RFC 5247.

The first research paper is, "A Practical Analysis of EAP Authentication Methods," by Alexandra Chiornita, Laura Gheorghe, and Daniel Rosner (Chiornita, et al). This paper reviews the different EAP methods used in the IEEE 802.1x Port Based Network Access Control comparing the most common types. These are listed in the table below:

Table 3 - Common EAP methods

Name	Code	Required/Optional	Developed by	Features
LEAP	17	Optional	Developed by Cisco, now standard	Cannot use certificates
EAP-MD5	4	Required	IEEE – RFC 3748	Cannot use certificates
EAP-TLS	13	Optional	IEEE – RFC 5216	Certificate-based
PEAP	25	Optional	Developed by Cisco, now standard	Certificate-based
EAP-PSK		Optional	IEEE – RFC 4764	Certificate-based
EAP-IKEv2		Optional	IEEE – RFC 5106	Certificate-based or password based
EAP-FAST	43	Optional	Developed by Cisco, now standard (RFC 4851)	Certificate-based
EAP-SIM	18	Optional	IEEE – RFC 4186	
EAP-AKA	23	Optional	IEEE – RFC 4187	

This paper reviews the impact of the three most common methods in EAP-MD5, EAP-TLS, and EAP-PEAP. The first one is mandatory for any device which implements EAP. One issue is that it offers minimum security, requiring only the supplicant to authenticate to the server. Bi-directional or mutual authentication is not supported. "Because it is based on MD5 hash function, is it vulnerable to dictionary and man-in-the-middle attacks and it does not support key generation [16]." The second EAP protocol reviewed in the paper is EAP TLS. While it is not required it does support certificates and provides for mutual authentication. The certificates it authenticates are normally to the device and not the user. One of the drawbacks is the infrastructure cost because it requires a certificate server as well as the authentication server. The third method reviewed in the paper is EAP-PEAP. This method has two stages, the first is the establishment of a tunnel between the supplicant and the authentication server. In the second stage, the actual exchange of authentication messages takes place. The paper then reviews the three methods using a criteria of authentication time, reauthentication time, packet loss during reauthentication and throughput. While throughput and packet loss shows no difference between the three methods, authentication time and reauthentication time did vary. In both cases MD5 had the least amount of time, averaging about 5 times smaller than the other two. TLS had the longest time on authentication. The difference in reauthentication was slight, less than 34ms average between the fastest method and the slowest. While the authors argue that this makes EAP-TLS less than satisfactory on a time sensitive network the difference is slight enough that a user would not find it significantly noticeable as the longest time delay would be .21 seconds.

In the second paper, “Research of AAA Messages based on 802.1x Authentication,” the team of Jiange Zhang, Yuanbo Guo, Yue Chen, Jun Ma (Zhang, Et al) constructed an AAA message for analysis. They do this using a client (supplicant), switch, and an authentication server, in their case using FreeRADIUS version 2.1.10. And while the rest of the paper is mostly a deconstruction of the messages in hexadecimal their conclusion is 802.1x authentication provides AAA and is effective in controlling user access to a network LAN. Yet it is their recommendation that can be of interest, “One improved method is authenticating twice. For example, it adopts identity authentication before network access authentication. Consequently, it will enhance the security of network [89].” I want to call attention to their recommendation because they suggest a double authentication, one for the identity user and the second for authentication to the network. This appears to suggest a separation of the user identity from the device identity. This can be critical and represents a vector of attack on the EAP, the authentication is completed and a new user without login credentials sits at the unattended workstation. While the paper used EAP MD5 which has no certificates, as we see from table 3 there are several other versions and these versions do have or require certificates.

EAP TLS is a version that uses certificates. Not only does TLS have certificates but it is bi-directional verification of identity. In other words, as a supplicant or peer how do I know that the server I am talking to in 802.1x or EAP is actually the server I should be talking to? How do I know the AP is the AP I wanted to talk to and not an evil twin? Fake APs, evil twins, are one of the attack vectors for the adversary. “Cybercriminals know that people regularly use public Wi-Fi and therefore they set up fake access points or compromise legitimate WiFi networks to intercept and manipulate their victim’s browsing. Their focus for

the attack is user's passwords, credit cards and other sensitive personal information [100]." One of the ways to counter this threat is by having mutual verification. EAP-TLS offers this in that not only the peer (supplicant) but also the server present's a certificate for verification. As you can see from the figure below, the server present's its certificate while requesting the client's certificate. This allows the client to verify it is talking to the server it wants to talk to. I wish to examine the exchange a little more in depth as well as EAP-TLS, remembering that one of our requirements is a bi-directional verification of identity.

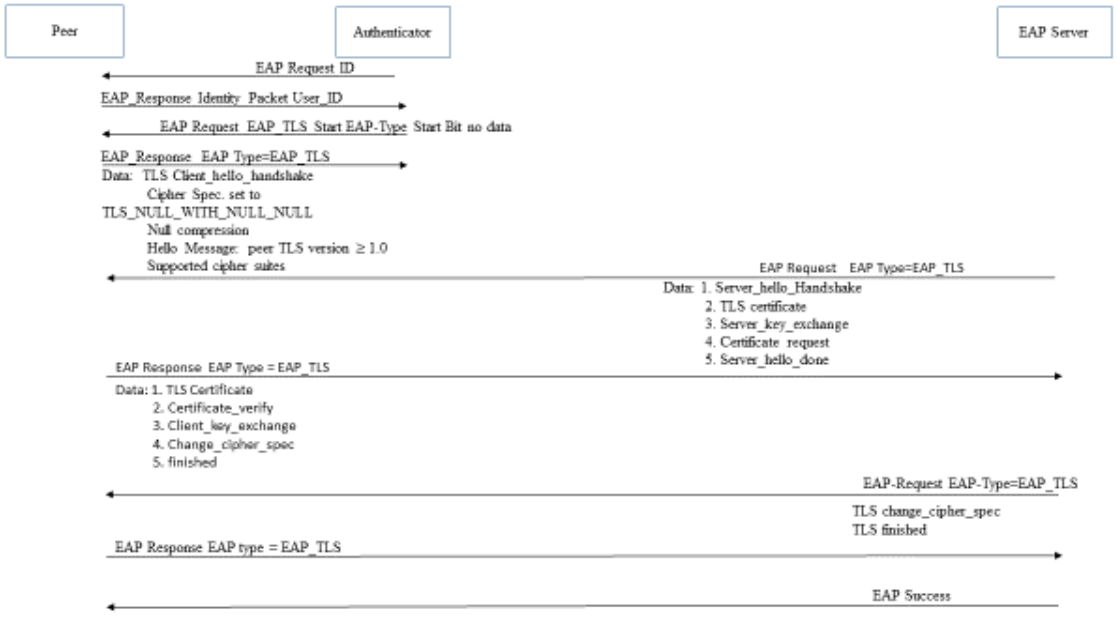

Figure 8 - TLS Authentication Exchange

The above figure documents the establishment of an EAP-TLS session as provided in RFC 5216. The session actors are the peer, the authenticator, and the EAP server. The authenticator can be an AP, it can be a switch, and thus EAP-TLS works not only on a wired network but also on a wireless 802.11 network. The client can be any device that can use 802.1x either wired or wireless. Thus, EAP-TLS is well-suited for the mixed device environment of a network. In EAP-TLS the peer begins the session by attempting to connect

at which point the AP or switch acting as an authenticator sends out a request, “Who are you?”, in effect asking for the client ID, EAP Request ID. The client says, “Ah ha, EAP” and sends an EAP Response with the client ID packet, which the authenticator passes on to the EAP server. From this point on the authenticator will act as a pass through for the various messages the client and EAP server generate. The EAP server send and EAP request with EAP-type set to TLS, the Start bit set and no data. The client says, I see you are not only speaking EAP but it is EAP TLS and so it sends an EAP response with EAP-Type set to TLS and a TLS client_hello_handshake. This hello handshake message has the TLS version number the client is using, a session ID, a random number, and the set of cipher suites the client supports. Now the EAP server sends an EAP-Request packet. In this packet the EAP-Type is set to TLS a server_hello_handshake message, the TLS certificate, the server_key_exchange, a certificate request, and then says, server_hello_done. The server hello message contains, the TLS version number, another random number, a session ID, and a cipher suite. The cipher suite is from the list the client sent. The client then says, EAP Response with EAP Type is EAP TLS, TLS certificate, certificate verify, a client key_exchange, change cipher specification, and a finished. Now the EAP server will verify the client certificate and digital signature then sends a EAP Request, EAP-Type set to EAP TLS and the change cipher followed by a TLS finish message. The client respond’s one last time in the handshake with EAP response, EAP-Type set to EAP TLS and the EAP server says EAP success. Of interest is the identity verification and the certificates as these can be tools in a mutual identity verification schema.

EAP is uniquely suited to provide not only user identity but also device identity. There are the two points within the EAP protocol regardless of the methodology you use for

identity. The first is the identity response. "The Identity Type is used to query the identity of the peer [85]." The second area is for those versions that use certificates. The certificates can contain extensive information but for my purpose it is sufficient to point out this part of RFC 3280, "The SubjectAltName MAY carry additional name types through the use of the otherName field. The format and semantics of the name are indicated through the OBJECT IDENTIFIER in the type-id field." Use of this field is not constrained to a specific type but can hold a variety of types, below is a small table listing those types:

Table 4 - RFC 3280 SubjectAltName Values

General Name	Choice	Identifier
otherName	[0]	OtherName,
rfc822Name	[1]	IA5String,
dNSName	[2]	IA5String,
x400Address	[3]	ORAddress,
directoryName	[4]	Name,
ediPartyName	[5]	EDIPartyName,
uniformResourceIdentifier	[6]	IA5String,
iPAddress	[7]	OCTET STRING,
registeredID	[8]	OBJECT IDENTIFIER

The use of this data type in the peer certificate can point toward a specific resource used to validate the identity of the device as well as its registered user.

3.5 Remaining issue

While authentication is an oft study field and many schemes exist for ensuring mutual authentication that mutual authentication is often device to device. In the authentication process while the peer consists of two parties the schemes do not make an effort to authenticate the user and the device. For authentication to fulfill its definition completely the entity in a computer system, composed of two separate pieces, the user and the device needs to be authenticated. Current practices are limited to authenticating the user and assuming the

device is a part of the user's identity. In some cases, EAP-TLS, IKE the peer device will be authenticated using a peer certificate however this then limits the authentication to the device possessing a certificate and while the certificate authority (CA) will make every effort to ensure the certificate represents an individual, there is no effort to associate the certificate to a specific device. This also occurs in the EAP server. Certificates are often transferred from server to server as hardware is upgraded. The solution proposed in Chapter 4 will address the issue of mutual authentication as well as ensure that the user and the device remain the same throughout each session, creating in effect true integrity in each transaction.

CHAPTER FOUR

PROPOSED SOLUTION

The attacks on networks and data stores are increasing daily. In their report, "Internet Security Threat 2016", Symantec stated, the following big numbers: "Over half a billion personal records were stolen or lost in 2015. With an average of 1.1 million web attacks blocked per day." In addition, they reported that smartphones and mobile devices are increasingly becoming the target of attacks. Just recently, one of the largest distributed denial of service attacks was launched against a target, DYN, which acts as a major player in the US market as a DNS provider bringing disrupting internet sites such as PayPal, Twitter, and Spotify. "What was most interesting about this attack was that it was largely carried out using an Internet of Things (IoT) botnet called Mirai (Linux.Gafgyt) [105]." The abuse of the common resource, the Internet and the hi-jacking of it as a highway for criminal intent continues to increase each year.

One of the very central features, privacy of the user is under attack not just through the use of the networks for criminal purposes, but country states on every continent are demanding unfettered access to user records. In China, they have established the Great Wall regularly blocking individuals with views the state would prefer to suppress. In the US, a place that generally regards itself as a bastion of right to self-expression and protection of individual privacy enshrined in its constitution, Federal, state, and local law enforcement agencies demand access providers give them records. The agencies have gone so far to use closed courts established by the Patriot Act of 2001 and its extensions to obtain these records. The NSA was caught in a recent scandal when a contractor revealed through leaked documents the existence of executive authorized federal programs. Privacy on the internet

has been revealed to be a sham for the average user. Facebook collects meta data and Google, one of the largest search engines on the internet if not the largest has been revealed to collect meta-data and using it to customize solutions selling that service to vendors and others.[2] As I previously reported in this paper there are three companies within the US that sell the data that not only identify devices but also their users on the internet. For the legal user, there is no privacy on the internet. As Daniel Newman stated in his article, "There is no privacy On the Internet of Things," he goes on to explain we, "want to play with the latest games, toys and widgets, but we by in large don't want to trade our cash for them. So instead we trade something else; our data and our privacy." Even more alarming for those who do desire privacy was the results of tests reported in, "FPDetective: dusting the web for fingerprinters," by Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel (Acar Et al) "we obtained the same results with respect to the number of fonts probed and other browser properties accessed, suggesting that Do Not Track (DNT) preferences are ignored by fingerprinters [5]." In another paper previous cited, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting, "Nikiforakis et al found, "Through our analysis, we discovered that, unfortunately, in all cases, the extensions were inadequately hiding the real identity of the browser, which could still be straightforwardly exposed through JavaScript." The team also found there were several major issues with the use of these anonymizing extensions. The old paradigm of an "internet for everyone and everyone for the internet," with a high value placed upon user privacy is a fallacy given the above statement of facts. It is time to make a paradigm shift. I

[2] http://www.pcworld.com/article/2986988/privacy/the-price-of-free-how-apple-facebook-microsoft-and-google-sell-you-to-advertisers.html is a good explanation of how the process works as well as a window into the scope of the information collected.

would like to suggest a new paradigm and in the process, show how device fingerprinting, can assist in the proper use of identification and authentication.

4.1 A New paradigm

In Elinor Ostrom's book, Governing the Commons, the Nobel Prize winner studies the governing of resources held in common. There is a suggestion of three models of governance of commonly held resources. The first is the case of leviathan, the use of the state and its structures to govern the commonly held resource. The second model is the use of privatization. While the case of the levitation has many proponents, including the current US government administration, as seen by this web page [107]. While proclaiming a victory for a free and fair Internet, the corollary of this statement is the "FCC just voted in favor of strong net neutrality rules" and infers strongly, the government has the power to regulate the Internet. There are, as Ostrom states in her book, drawbacks[3] to this method. One of the drawbacks is a central agency is divorced from the behavior of the primary users and therefore will incorrectly pose sanctions. A second drawback is no central agency has all the information, correctly sourced, understood, and current. Often government agency information is outdated or fails to account correctly for user behavior.

The second model is the privatization of the commonly held resource. Doing so by creating a system of private property rights, as has been suggested in EU decisions and opinions in regard to Directive 2002/58/EC which states the users are the owners of their data on the Internet. This moves us down the road toward interlocking property rights over data on the internet, access to, use of applications, and services obtained through the internet.

[3] The discussion of central agency is found on pages 8-11 of the sourced book. It makes heavy use of game theory to project behavior of the governed.

There is a third model in Ostrom's book and one that should be of interest since in truth the internet and networks in general meet a definition of a common resource. As Investopedia is kind enough to point out, "A common resource is a resource, such as water or pasture, which provides users with tangible benefits. A major concern with common resources is overuse, especially when there are poor social-management systems in place to protect the core resource."[4] This model is one in which the users would make a binding contract for a cooperative strategy in which the users would determine a strategy for providing a penalty to the non-cooperative members. In this case, the non-cooperative members are those who misappropriate the internet resources.

Regardless of which model is chosen, the days of anonymity across the web are coming to an end. If not through rules and regulation, then through market forces as personal data and user behavior have been defined as marketable information. Therefore, there needs to be a paradigm shift. Instead of a paradigm where users' privacy is protected we recognize what has already happened, user transactional privacy should be protected but efforts to protect user identity from discovery, ineffective in the majority of instances should be modified to end this effort.

Instead of if your device has a connector and the user is authorized, track all devices. In the early years of the automobile industry anyone with the right equipment and tools could and did build automobiles for use and sale. However, that began to change in 1954 with the first use of standardized vehicle identification numbers VIN). As the industry matured, VINs were used for a variety of purposes. Today at the website for National Highway Traffic and

[4] I would argue this definition is especially true of the Internet and networks in general. However, in place of overuse we have misuse. This misuse would be the use of the network to gain unauthorized economic advantage through the blocking of services, the unauthorized access to personal information, and unauthorized use of shared resources.

Safety Administration you can, using a VIN, determine if the auto is subject to recall, has it been reported stolen, check your vehicle's history including ownership and accident record. The automobile is used for a variety of purposes many related to accomplishing life tasks using a network of highways. Computers these days are also used to accomplish many of life's tasks using a network of connecting electronic pipes maintained by a variety of sources, ranging from the private to public (non-profit) to government owned networks. As part of the paradigm shift, devices would be identified by a unique ID. This unique ID would be permanently associated with the device. The unique ID will be used to authenticate and reauthenticate the device to the network. A backend data file of the unique ID will be associated with a user/ownership record. The whole process of identification and authentication will change to look more like an EAP authentication with certificates.

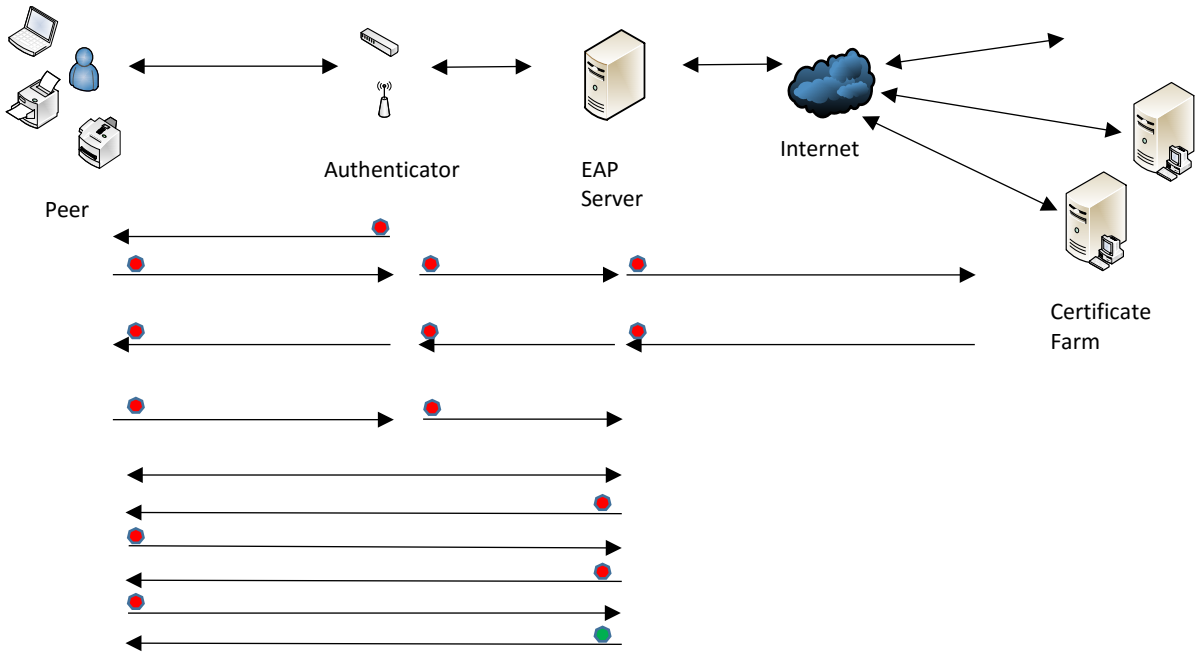

Figure 9 - Modified EAP process

There is one new actor in the extended process, this actor is the set of certificate servers which act as a farm or storage location. Let's take a look at the infra-structure first. As in

EAP you have three sets of actors, the peer, the authenticator, and the EAP server. In EAP, 802.1x implementation you will have a server which maintains information on the devices using which port. Searching its manually populated database the EAP server will notify the authenticator if the device is allowed to connect to the network. Normally the EAP server will store user in a flat data file or it can refer to an external server containing SQL, Kerberos, LDAP, or Active Directory for credentials. In extended EAP, there is an additional source of information, the certificate farm (CF). The contents of the certificate farm are similar to what an EAP server contains with a slight modification. The slight modification is shown in the figure below:

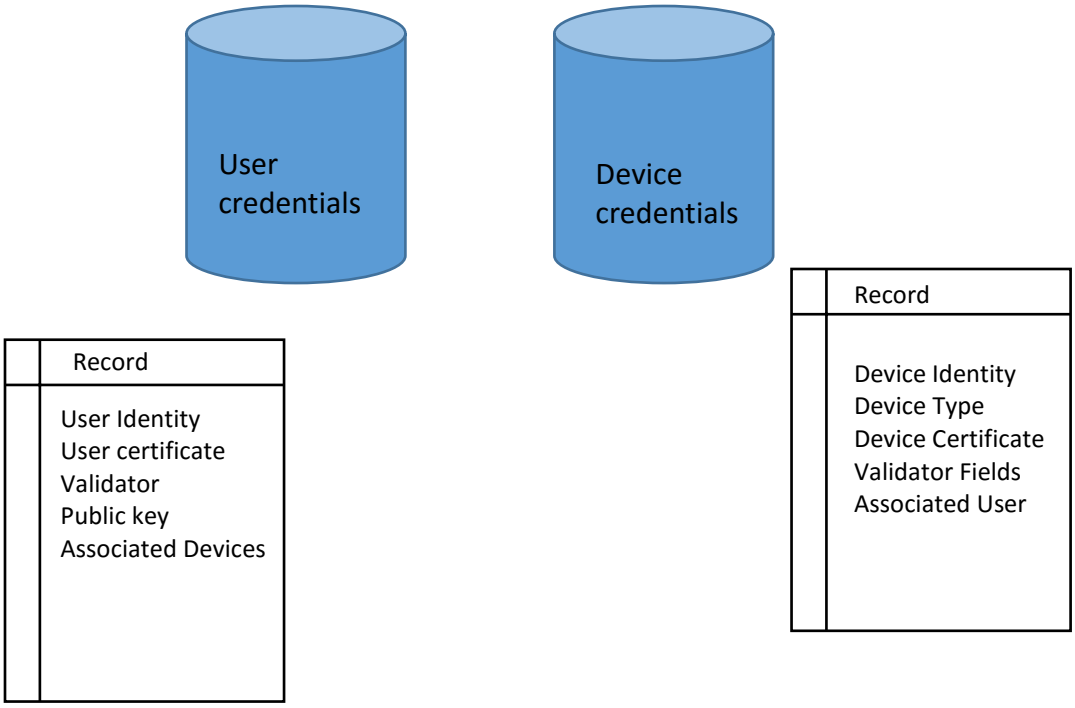

Figure 10 - Certificate Farm Data Files

The certificate farm has two databases. The first database is a relational database and contains the following files, a user file, a user certificate file, and a user associated device file. The user file will have the user identity and a unique key index. The user certificate

file will have the unique key index, related to the unique key index of the user file, the user certificate which contains the validator, and the user's public key. The last related data file will be the associated device file. The associated device file will have the unique key index related to the key index from the user file. This file will allow multiple devices to be associated with a particular user. The second database in the certificate farm is also a relational database composed of the following data files; device identity, device certificate, device validators, and associated users. The device identity file will contain fields for device identity, device type, device unique descriptor, and a unique device id index field. The device certificate file will contain the device id index field, the device certificate. The third file will contain the unique device id field related to the device identity file and a set of validator fields. The last file in the relational data base is a record of devices from the first file and contains a list of associated users with the device id from the first file. This unique id is not unique in this file.

In order to understand how this works, I want to use the following use cases; the first is the user with a device will authenticate through EAP. The second use case will be a device will authenticate. The second use case describes how the internet of things (IoT) as well as devices not traditionally recognized in a network access control (NAC) process will authenticate and of course re-authenticate as needed.

The first use case, designated 1.0, has multiple sub-cases. In the first sub-case, designated 1.1, the user is a member of an organization. In the second sub-case, designated 1.2, the user is a person wishing to use a public AP. In the third sub-case, designated 1.3, a person wishes to use a public network. In use case 1.1, a user approaches the network and wants to connect. In today's workplace, the device might be an already constrained device

such as a company owned laptop/workstation/mobile device or it might be a bring your own device (BYOD) smartphone, laptop, or tablet. Regardless of the device and the user the following are already pre-existing, the user has a certificate and so does the device. For use case 1.1 the existence of the user certificate is because the company has granted him a certificate to use while logging into the network. These credentials are stored in the EAP server and also stored in the certificate farm. In the case of the device the device certificate may reside on the EAP server but it definitely is available in the certificate farm, in the device database. The validators for the device in the device database are those items that are used to construct the device's fingerprint. These validators are items that based upon the device are passively observable. For the laptop/workstation it will consist of the clock skew of the processor measured against the CF's device server. Additional validator fields will include the MAC address, the OS, and the browser unique identifiers. For a smartphone/data phone/tablets the validator fields will be the accelerator skew, the clock skew, and the MAC identifier. For a sensor, the validator fields will consist of clock skew, MAC, and location.

The peer connects or powers on the device. The first EAP Request flows from the authenticator, EAP request for client ID. The first of two messages flow back, EAP response with peer user ID, P_u. The second response message is peer device ID, P_d. The authenticator passes the messages to the EAP server. The EAP server reviews its user and device records looking for a match, M, $M = P_u + P_d$. If the EAP server has a match, there is no need to send a request to the CF. If there is no match either for P_u or for P_d or for both the EAP server queries the CF for matches. The CF will search for either both values or for only the value the EAP server sends to it. If it recovers a match, the information is provided to the EAP server. Now the EAP server will add the records to its file and the EAP process

continues with the EAP Request EAP TLS start bit set. The EAP response client Hello handshake continues as before. The next change comes in the step where the EAP server requests the certificate. In this step both P_u and P_d will provide their certificates for validation. The server will validate both certificates and on success send EAP request EAP type = EAP-TLS change cipher spec and TLS finished. P_u and P_d respond with EAP response followed by the EAP server's EAP success message. In this manner both P_u and P_d are identified and authenticated to the network, A or $A = (P_{ui}P_{uc})(P_{di}P_{dc})$ where P_{ui} is Peer user identity, P_{uc} is the Peer user certificate and P_{di} is Peer device identity, and P_{dc} is the Peer device certificate. What happens if there is no match at the EAP server or the CF? Remember in this use case both P_u and P_d and must be authenticated. It is not sufficient for one to be validated without the other being validated. Thus, if either of the two elements are invalid then the EAP server refuses connection.

In use case 1.2, the user is a person wishing to use a public AP, whether this AP is in a hotel wired or a wireless AP there is an assumption the EAP server does not exist. As such the credentials or certificates do not exist on the AP. Therefore, all the transactions will occur with the CF. The peer connects or powers on the device. The first EAP Request flows from the authenticator, EAP request for client ID. The first of two messages flow back, EAP response with peer user ID, P_u. The second response message is peer device ID, P_d. The authenticator passes the messages to the CF. The CF reviews its user and databases looking for a match, M, $M = P_u + P_d$. If it recovers a match, the information is provided to the EAP authenticator sending the EAP Request EAP TLS start bit set. The EAP response client Hello handshake continues as before. In the place where the EAP server would send its certificate the CF having the certificate for the AP provides that to the peer. The next change

comes in the step where the EAP server requests the certificate. In this step both P_u and P_d will provide their certificates for validation. The CF will validate both certificates and on success send EAP request EAP type = EAP-TLS change cipher spec and TLS finished. P_u and P_d respond with EAP response followed by the EAP server's EAP success message. In this manner both P_u and P_d are identified and authenticated to the network, A or $A = (P_{ui}P_{uc})(P_{di}P_{dc})$ where P_{ui} is Peer user identity, P_{uc} is the Peer user certificate and P_{di} is Peer device identity, and P_{dc} is the Peer device certificate. What happens if there is no match in the CF? Remember in this use case both P_u and P_d and must be authenticated. It is not sufficient for one to be validated without the other being validated. Thus, if either of the two elements are invalid then the CF tells the authenticator to refuse the connection.

The last use case is the peer wishes to use a public network. This would be a network that is provided not just as an access point but also has resources attached to it, normally this type of network is associated with a library or similar community service and has a firewall isolating its public portion from the business services of the community service. In this use case the process is exactly the same as in use case 1.1. The difference would be the timed storage of credentials would periodically expire so a heavier use of the CF will occur.

Moving to use case 2 which is where the peer is a device only. You normally find this type of use case where supervisory control and data acquisition (SCADA) system or sensor resides on the network. In the internet of things (IoT) such devices range from the SCADA mentioned above to the refrigerator sensing what you use and sending an order to the grocery store or even the control system for your DVD-R or heating or lights, or simply a web camera sending images to a backend storage device as part of a building security system. Sense one of our goals is to not have any unknown devices attached to the network our EAP-

TLS process has to account for these devices. The major problem here is EAP-TLS is a request response system. Some of the devices contain a processor who's only set of instructions are to check a sensor and send the results to a monitoring station. The recent DDoS attack on DYN proved that these devices must be accounted for in any network security plan. If you remember the certificate for each device is stored in the CF and associated with a record for the device with specific identifiers, validators. These validators are items that can be passively observed in the network traffic of a device. We now have a process to cover this type of devices connecting to the network with periodic reauthentication that looks similar to the process described below:

Step one the device attempts to connect, and the authenticator sends the EAP Request ID. Normally the peer sends back its user ID but in this case, the EAP response will send a user ID message with additional information. This additional information is, device type, current time, MAC, location, and associated user set to the business/personal entity deploying the device. The EAP server will follow the protocol in search its records and if found will open the port, if not found it will query the CF. If the CF finds a match in its device database, then it will send the records to the EAP server for inclusion in its validation data file. The EAP server will then open the port. If there is no match in the EAP server or in the CF, then the port will remain closed.

As you can see this is a paradigm shift requiring every device on the network is accounted for and assigned to an owner. This also does away with user anonymity and assigns a particular device or set of devices to each user. This process also extends the reach of the certificate servers because sites or servers or user will not be the only items with a certificate but each user and device will have a certificate. There are other consequences of

making this shift but are beyond the scope of this current paper other than to note that this scheme does not fall afoul of the EU privacy rule due to the exceptions included in the opinion, "As described in Opinion 04/2012, Article 5(3) allows for processing to be exempt from the requirement of consent, if one of the following criteria is satisfied:
CRITERION A: technical storage or access 'for the sole purpose of carrying out the transmission of a communication over an electronic communications network'.
CRITERION B: technical storage or access which is 'strictly necessary in order for the provider of an information society service explicitly requested by the subscriber or user to provide the service' [93]." This process changes the way in which access is provided to the user or subscriber and the information gathered would be necessary to provide service to the user thus meeting criterion B of the opinion.

4.2 Creating the device fingerprint for the certificate farm

There are three different stages in the creation and use of the device fingerprinting in authentication. These stages are generically speaking: pattern or profile generation, enrollment, and validation. During pattern generation the pattern used for the fingerprint is created and assigned to a profile. This profile is stored and assigned an arbitrary identifier. The focus during this stage is to clearly define a pattern that can be reviewed and tested against. In stage two of the process, the enrollment stage, the pattern has been confirmed as existing and distinct enough to use in measurement. It is transferred to the enrollment container for use against a trace. During the enrollment stage the pattern is validated as unique and not pre-existing by reviewing previously existing profiles. If there is no pre-existing profile with the same pattern then the profile is transferred to the validation stage.

At this point the profile is viewed against a trace of the data set including the device. If the accuracy of identity meets the pre-determined threshold, the profile is transferred to the certification farm for pre-staging. Pre-staging is the device has not been seen on the network. Once the device is seen on the network, the balance of the record is complete and the record if is stored with a certificate containing all the required information for use in subsequent identification and authentication efforts. It is this three phased process that can be accomplished in a pre-sale environment by the manufacturer.

The device fingerprints once created are registered to the Certificate Farm. Similarly, to the automobile industry, this can be done from the manufacturer's facility. The initial record for each device manufactured for sale would be forwarded to the Certificate Farm (CF) or available from the manufacturer's site to be downloaded by the CF as needed. Upon first use the CF would begin by populating the rest of the record in their data file and the certificate would be complete and available for authentication. By having the manufacturer capture the data you will have a controlled environment which will assist capturing critical data. And in the case of the smartphone/data phone/tablets will easily capture the accelerator data critical to the creation of the device digital fingerprint. It will also ensure the MAC address is not spoofed in the device fingerprint. Finally, as all of the information in the device fingerprints are captured through passive observation it would eliminate this stage of the process.

There are so many ways to identify devices, to create device fingerprints. In chapter two we lightly surveyed four major ways devices are fingerprinted today, hardware including clock skew, software, user behavior, and the browser. In researching for this paper there was one team that suggested the unique device fingerprint be associated with a charging cable

[15]. The object is to strengthen security in a poly device network which includes not only the user/laptop hybrids of traditional networks but also sensors, SCADA, smart devices, and various appliances. This objective is executed in the proposed solution because in the network every device is identified through the authentication process. This is accomplished by uniquely crafting an identity for a device based upon several characteristics which can be used to validate their identity through passive observation and without requiring input.

In constructing the device fingerprint for the laptop/workstation the clock skew C_d measured against the clock of the CF device storage F_v, the OS or O_d , the MAC (which can be spoofed) or A_d and the browser identifiers (B_1, B_2, B_3, and B_4) are used to construct the device fingerprint. The browser identifiers are contained in the table below:

Table 5 - Browser Features in Fingerprint

Feature	Vector measured	Identifier
Browser type(s)	HTTP	B_1
Time of day	HTTP	B_2
Country	HTTP	B_3
Fonts	Script	B_4

The laptop/workstation device fingerprint is stated thus:

$$F = ((C'_d(t) - F'_v(t)) + O_d + A_d + B_1 + B_2 + B_3 + B_4).$$

The fingerprint for a smartphone/data phone/tablet would be constructed using the accelerator skew from true ε_i, the clock skew using the CF server as the device receiving, and the MAC resulting in fingerprint looking something like this:

$$F = ((C'_d(t) - F'_v(t)) + \varepsilon_i + A_d).$$

The simple sensor set for SCADA or for IP security cameras with clock skew, location W_d will yield a fingerprint looking like this: $F = ((C'_d(t) - F'_v(t)) + W_d + A_d)$. By adding together various vectors to constitute the fingerprint the opportunity for uniqueness increases.

In this section we have looked at the process used to create a device fingerprint and based upon the research surveyed in chapter two of this paper suggested a composite device fingerprint that makes use of the clock skew in all three sets of devices.

CHAPTER 5

SUMMARY

This paper explored the proposition, device finger printing can be used as part of a multi-factor identification/authentication scheme that adds security to a network. It advanced the concept of not only identifying the user for access to the network but, the device and user form a combination that should be identified prior to network authorization. Its goal was accomplished by looking at relevant research and where possible the data sets that supported that research. The terms of identification and authentication were explored to ensure the issue was understood and the current definition was refined to add the concept of both entities presenting an identity for authentication. The paper presented a concept to included methods for handling those devices sensors, printers, etc. that do not participate in a bi-directional handshake. A process was offered for providing access to the networks using the EAP-TLS framework as a touchline. Finally, sample of how the device fingerprint for the various categories that make up the device population in a poly device network was proposed as well as how to populate the certificate farm.

REFERENCES

[1] U. A. Abdurrahman, M. Kaiiali, and J. Muhammad, “A new mobile-based multi-factor authentication scheme using pre-shared number, GPS location and time stamp,” in *Electronics, Computer and Computation (ICECCO), 2013 International Conference on*, 2013, pp. 293–296.

[2] B. Aboba, H. Levkowetz, D. Simon, and P. Eronen, “RFC 5247 Extensible Authentication Protocol (EAP) key management framework,” 2008.

[3] K. A. Abu Bakar and G. R. Haron, “Adaptive authentication based on analysis of user behavior,” in *Science and Information Conference (SAI), 2014*, 2014, pp. 601–606.

[4] G. Acar *et al.*, “FPDetective: dusting the web for fingerprinters,” 2013, pp. 1129–1140.

[5] G. Acar, “Obfuscation for and against device fingerprinting Position Paper for Symposium on Obfuscation New York University, February 15, 2014.”

[6] M. M. Althobaiti and P. Mayhew, “Usable security of authentication process: New approach and practical assessment,” in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2015, pp. 179–180.

[7] K. Altinkemer and T. Wang, “Cost and benefit analysis of authentication systems,” *Decision Support Systems*, vol. 51, pp. 394–404, Jan. 2011.

[8] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, “On the reliability of wireless fingerprinting using clock skews,” 2010, p. 169.

[9] R. K. Banyal, P. Jain, and V. K. Jain, “Multi-factor Authentication Framework for Cloud Computing,” in *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation (CIMSim)*, 2013, pp. 105–110.

[10] Block, Dimitri, Hendrik Fliedner, Niels, and Meier, Uwe, “CRAWDAD dataset init/factory (v. 2016-06-13).” Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD), 2016.

[11] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, “Mobile Device Identification via Sensor Fingerprinting,” *arXiv preprint arXiv:1408.1416*, 2014.

[12] T. Borgohain, A. Borgohain, U. Kumar, and S. Sanyal, “Authentication Systems in Internet of Things,” *International Journal of Advanced Networking & Applications*, vol. 6, no. 4, pp. 2422–2426, Jan. 2015.

[13] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles, “Active behavioral fingerprinting of wireless devices,” in *Proceedings of the first ACM conference on Wireless network security*, 2008, pp. 56–61.

[14] W. E. Burr, *et al.*, “NIST SP 800-63-2 Electronic Authentication Guideline,” *NIST SP 800-63-2*. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf. [Accessed: 14-Jun-2016].

[15] A. C.-F. Chan and J. Zhou, “Cyber-Physical Device Authentication for the Smart Grid Electric Vehicle Ecosystem,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 7, pp. 1509–1517, Jul. 2014.

[16] A. Chiorniţă, L. Gheorghe, and D. Rosner, “A practical analysis of EAP authentication methods,” in *9th RoEduNet IEEE International Conference*, 2010, pp. 31–35.

[17] S. Choi and D. Zage, “Addressing insider threat using ‘Where you are’ as fourth factor authentication,” in *2012 IEEE International Carnahan Conference on Security Technology (ICCST)*, 2012, pp. 147–153.
[18] B. Danev, D. Zanetti, and S. Capkun, “On Physical-Layer Identification of Wireless Devices,” *ACM Computing Surveys*, vol. 45, no. 1, p. 6:1-6:29, Nov. 2012.
[19] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee, “Identifying unique devices through wireless fingerprinting,” in *Proceedings of the first ACM conference on Wireless network security*, 2008, pp. 46–55.
[20] T. Dierks and C. Allen, “RFC 2246 - The TLS protocol version 1.0,” 1999.
[21] E. dos Santos, J. E. Martina, and R. F. Custodio, “Towards a Formal Verification of a Multi-factor Authentication Protocol Using Automated Theorem Provers,” in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012, pp. 84–91.
[22] P. Eckersley, “How unique is your web browser?,” in *International Symposium on Privacy Enhancing Technologies Symposium*, 2010, pp. 1–18.
[23] J. P. Ellch, “Fingerprinting 802.11 Devices,” DTIC Document, 2006.
[24] C.-I. Fan and Y.-H. Lin, “Provably Secure Remote Truly Three-Factor Authentication Scheme With Privacy Protection on Biometrics,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 933–945, Dec. 2009.
[25] D. B. Faria and D. R. Cheriton, “Detecting identity-based attacks in wireless networks using signalprints,” in *Proceedings of the 5th ACM workshop on Wireless security*, 2006, pp. 43–52.
[26] R. Fink, “A Statistical Approach to Remote Physical Device Fingerprinting,” in *IEEE Military Communications Conference, 2007. MILCOM 2007*, 2007, pp. 1–7.
[27] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker, “Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting,” in *Usenix Security*, 2006.
[28] 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)*, 2010, pp. 383–392.
[29] S. Gibbs, “Europe’s next privacy war is with websites silently tracking users,” *the Guardian*. [Online]. Available: http://www.theguardian.com/technology/2014/nov/28/europe-privacy-war-websites-silently-tracking-users. [Accessed: 21-May-2015].
[30] L. G. Greenwald and T. J. Thomas, “Understanding and preventing network device fingerprinting,” *Bell Labs Technical Journal*, vol. 12, no. 3, pp. 149–166, Fall 2007.
[31] F. A. Guenane and G. Pujolle, “Strong virtual network authentication using EAP-TLS smart-cards,” in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, 2012, pp. 197–199.
[32] V. Hazlewood, P. Kovatch, M. Ezell, M. Johnson, and P. Redd, “Improved Grid Security Posture through Multi-factor Authentication,” in *2011 12th IEEE/ACM International Conference on Grid Computing (GRID)*, 2011, pp. 106–113.
[33] R. Housley and B. Aboba, “RFC 4962 Guidance for authentication, authorization, and accounting (AAA) key management,” 2007.
[34] D.-J. Huang, K.-T. Yang, C.-C. Ni, W.-C. Teng, T.-R. Hsiang, and Y.-J. Lee, “Clock Skew Based Client Device Identification in Cloud Environments,” 2012, pp. 526–533.

[35] D.-J. Huang, K.-T. Yang, W.-C. Teng, and G.-M. Chiu, “Design of client device identification by clock skew in clouds,” in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 1133–1138.

[36] W. Huang and R. Li, “WLAN Authentication System Based on the Improved EAP-TLS Protocol,” College of Information and Electrical Engineering Hebei University of Engineering, 2015.

[37] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, “Robust Multi-Factor Authentication for Fragile Communications,” *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, Nov. 2014.

[38] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks timing and synchronization for time-sensitive applications in bridged local area networks*. New York: Institute of Electrical and Electronics Engineers, 2011.

[39] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks media access control (MAC) service definition*. New York: Institute of Electrical and Electronics Engineers, 2012.

[40] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks: timing and synchronization for time-sensitive applications in bridged local area networks. Corrigendum 1, Corrigendum 1*,. 2013.

[41] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks: overview and architecture*. 2014.

[42] Institute of Electrical and Electronics Engineers, IEEE Computer Society, LAN/MAN Standards Committee, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks port-based network access control*. New York: Institute of Electrical and Electronics Engineers, 2010.

[43] Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks: secure device identity*. New York: Institute of Electrical and Electronics Engineers, 2009.

[44] Y. Ishikawa, N. Yamai, K. Okayama, and M. Nakamura, “An Identification Method of PCs behind NAT Router with Proxy Authentication on HTTP Communication,” 2011, pp. 445–450.

[45] H. Jagadeesan and M. S. Hsiao, “A novel approach to design of user re-authentication systems,” in *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, 2009. BTAS ’09*, 2009, pp. 1–6.

[46] S. Jana and S. K. Kasera, “On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 449–462, Mar. 2010.

[47] S. Jeon, “Four-factor verification methodology for entity authentication assurance,” in *2011 International Conference on Information Science and Applications (ICISA)*, 2011, pp. 1–4.

[48] M. S. Johns, “RFC 1413 - Identification protocol,” 1993.

[49] A. Joshi, S. Kumar, and R. H. Goudar, “A More Multifactor Secure Authentication Scheme Based on Graphical Authentication,” in *2012 International Conference on Advances in Computing and Communications (ICACC)*, 2012, pp. 186–189.

[50] T. M. Jurgensen and S. B. Guthery, “Smart cards: the developer’s toolkit,” presented at the Conference on Network and Information Systems Security, 2011, pp. 1–6.

[51] G. Kathrine and E. Kirubakaran, “Four-Factor based Privacy Preserving Biometric Authentication and Authorization Scheme for Enhancing Grid Security,” *International Journal of Computer Applications*, vol. 30, no. 5, 2011.

[52] T. Kohno, A. Broido, and K. C. Claffy, “Remote physical device fingerprinting,” *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, Apr. 2005.

[53] M. Kotha, M. Singirikonda, M. Nirosha, and G. Manjunath, “A Unique Wireless Device Fingerprinting Technique for Secured Data Communication in Wireless Network.” International Journal of Computer Applications, Apr-2012.

[54] F. Lanze, A. Panchenko, B. Braatz, and A. Zinnen, “Clock skew based remote device fingerprinting demystified,” in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 813–819.

[55] J.-C. Liou, G. Egan, J. K. Patel, and S. Bhashyam, “A Sophisticated RFID Application on Multi-Factor Authentication,” in *2011 Eighth International Conference on Information Technology: New Generations (ITNG)*, 2011, pp. 180–185.

[56] J. Martin, A. Rajan, and B. Steigerwald, “Authenticate Once and Be Done: User-Centric Authentication Through Rich Device Capabilities,” *Intel Technology Journal*, vol. 18, no. 4, pp. 8–28, Nov. 2014.

[57] M. M. Mohammed and M. Elsadig, “A multi-layer of multi factors authentication model for online banking services,” in *2013 International Conference on Computing, Electrical and Electronics Engineering (ICCEEE)*, 2013, pp. 220–224.

[58] H. H. Mojtaba Alizadeh, “Feasibility of Implementing Multi-factor Authentication Schemes in Mobile Cloud Computing,” 2014.

[59] S. Mondal and P. Bours, “Combining keystroke and mouse dynamics for continuous user authentication and identification,” in *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, 2016, pp. 1–8.

[60] S. J. Murdoch, “Hot or not: Revealing hidden services by their clock skew,” in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 27–36.

[61] C. Neumann, O. Heen, and S. Onno, “An Empirical Study of Passive 802.11 Device Fingerprinting,” in *2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2012, pp. 593–602.

[62] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, “Device fingerprinting to enhance wireless security using nonparametric Bayesian method,” in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1404–1412.

[63] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, “Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting,” in *2013 IEEE Symposium on Security and Privacy (SP)*, 2013, pp. 541–555.

[64] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, “On the Workings and Current Practices of Web-Based Device Fingerprinting,” *IEEE Security Privacy*, vol. 12, no. 3, pp. 28–36, May 2014.

[65] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*, Unknown edition. Cambridge ; New York: Cambridge University Press, 1990.

[66] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 99–110.

[67] C. Pavlovski, C. Warwar, B. Paskin, and G. Chan, "Unified framework for multifactor authentication," in *Telecommunications (ICT), 2015 22nd International Conference on*, 2015, pp. 209–213.

[68] L. Polcak, J. Jirasek, and P. Matousek, "Comment on 'Remote Physical Device Fingerprinting,'" *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 494–496, Sep. 2014.

[69] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "GTID: A Technique for Physical Device and Device Type Fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2014.

[70] B. Ramsey, B. Mullins, M. Temple, and M. Grimaila, "Wireless Intrusion Detection and Device Fingerprinting through Preamble Manipulation," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2014.

[71] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers," *Journal of Computer and System Sciences*, vol. 80, pp. 591–601, May 2014.

[72] S. U. Shah, F. Fazl-e-Hadi, and A. A. Minhas, "New Factor of Authentication: Something You Process," in *International Conference on Future Computer and Communication, 2009. ICFCC 2009*, 2009, pp. 102–106.

[73] Y. Shah, V. Choyi, and L. Subramanian, "Multi-factor Authentication as a Service," 2015, pp. 144–150.

[74] S. Sharma, H. Saran, and S. Bansal, "An empirical study of clock skew behavior in modern mobile and hand-held devices," in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, 2011, pp. 1–4.

[75] B. Sieka, "Active fingerprinting of 802.11 devices by timing analysis," in *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, 2006, vol. 1, pp. 15–19.

[76] D. Simon, B. Aboba, and R. Hurst, "RFC 5216 - The EAP-TLS authentication protocol," 2008.

[77] N. M. Smith, M. Sheller, and N. Heldt-Sheller, "Adding Nontraditional Authentication to Android," *Intel Technology Journal*, vol. 18, no. 4, pp. 120–137, Nov. 2014.

[78] T. Suen and A. Yasinsac, "Ad hoc network security: Peer identification and authentication using signal properties," in *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, 2005, pp. 432–433.

[79] Y. Tang, "A User Authentication Protocol based on Multiple Factors," *Journal of Networks*, vol. 9, no. 10, Oct. 2014.

[80] D. Todorov, *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*, 1 edition. Boca Raton: Auerbach Publications, 2007.

[81] A. S. Uluagac, W. Liu, and R. Beyah, "A multi-factor re-authentication framework with user privacy," in *2014 IEEE Conference on Communications and Network Security (CNS)*, 2014, pp. 504–505.

[82] A. S. Uluagac, S. V. Radhakrishnan, C. Corbett, A. Baca, and R. Beyah, “A passive technique for fingerprinting wireless devices with Wired-side Observations,” in *2013 IEEE Conference on Communications and Network Security (CNS)*, 2013, pp. 305–313.

[83] J. A. A. J. Valentino-DeVries, “Race Is On to ‘Fingerprint’ Phones, PCs,” *Wall Street Journal*, 30-Nov-2010.

[84] J. A. Vila, J. Serna, M. Medina, and A. Sfakianakis, “An Analysis of n-factor Authentication in e-Banking Environments,” *Journal of Information Assurance & Security*, vol. 9, no. 2, pp. 104–117, Apr. 2014.

[85] J. R. Vollbrecht, B. Aboba, L. J. Blunk, H. Levkowetz, and J. Carlson, “RFC 3748 - Extensible Authentication Protocol (EAP).” [Online]. Available: https://tools.ietf.org/html/rfc3748. [Accessed: 23-Jun-2016].

[86] M. A. Wright, “RFC 3280 - A look at public key certificates,” *Network Security*, vol. 1998, no. 2, pp. 10–13, 1998.

[87] Q. Xu, R. Zheng, W. Saad, and Z. Han, “Device Fingerprinting in Wireless Networks: Challenges and Opportunities,” *arXiv:1501.01367 [cs]*, Jan. 2015.

[88] J. Yu, G. Wang, Y. Mu, and W. Gao, “An Efficient Generic Framework for Three-Factor Authentication With Provably Secure Instantiation,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2302–2313, Dec. 2014.

[89] J. Zhang, Y. Guo, Y. Chen, and J. Ma, “Research of AAA messages Based on 802.1 x authentication,” in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2015, pp. 618–621.

[90] “Security: Identification, Authentication, and Authorization,” *danielmiessler.com*, 04-Oct-2005. .

[91] “Four-factor Authentication,” *Dustin D. Trammell*, 2008. .

[92] “Common Resource,” *Investopedia*, 22-Nov-2009. [Online]. Available: http://www.investopedia.com/terms/c/common-resource.asp. [Accessed: 31-Oct-2016].

[93] “Article 29 Data Protection Working Party: Opinion 9/2014 on the application of Directive 2002/58/EC to device fingerprinting.”

[94] “Customer Authentication and Fraud Prevention Solutions | iovation.” [Online]. Available: https://www.iovation.com/. [Accessed: 16-Aug-2016].

[95] “CWSP- EAP TLS | mrn-cciew.” [Online]. Available: https://mrncciew.com/2014/08/26/cwsp-eap-tls/. [Accessed: 27-Oct-2016].

[96] “Device-Fingerprinting-and-Online-Fraud-Protection-Whitepaper.pdf.” .

[97] “Device Fingerprinting Methodology,” *MobileAppTracking Support*. [Online]. Available: http://support.mobileapptracking.com/entries/21771055-Device-Fingerprinting-Methodology. [Accessed: 21-May-2015].

[98] “Difference between authentication and identification [Crypto and Security perspective] - Information Security Stack Exchange.” [Online]. Available: http://security.stackexchange.com/questions/10933/difference-between-authentication-and-identification-crypto-and-security-perspe. [Accessed: 14-Jun-2016].

[99] “Difference Between Identification & Authentication.” [Online]. Available: http://science.opposingviews.com/difference-between-identification-authentication-3471.html. [Accessed: 14-Jun-2016].

[100] “Don’t bank on public Wi-Fi being secure: 3-in-4 hotspots vulnerable, claim experts.” [Online]. Available: https://www.broadbandgenie.co.uk/blog/20160630-public-wifi-hotspots-vulnerable. [Accessed: 28-Oct-2016].
[101] “How Fraudsters Are Disguising PCs to Fool Device Fingerprinting,” *Security Intelligence*. .
[102] “How NAT Works.” [Online]. Available: https://technet.microsoft.com/en-us/library/cc756722(v=ws.10).aspx. [Accessed: 14-Oct-2016].
[103] “Information Management March/April 2015 Page 26.” [Online]. Available: http://imm.arma.org/publication/index.php?i=247767&pjs=1&p=32&pn=&ver=flex&pid=. [Accessed: 20-May-2015].
[104] “Internet Security Threat Report 2016, Vol 21, April 2016.”
[105] “Mirai: what you need to know about the botnet behind recent major DDoS attacks,” *Symantec Security Response*. [Online]. Available: http://www.symantec.com/connect/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-attacks. [Accessed: 30-Oct-2016].
[106] “Multi-Layer Device Fingerprinting.” .
[107] “Net Neutrality: A Free and Open Internet | The White House.” [Online]. Available: https://www.whitehouse.gov/net-neutrality. [Accessed: 31-Oct-2016].
[108] “NISP SP 1800-2a - Identity and Access Management, Executive Summary.” [Online]. Available: https://nccoe.nist.gov/sites/default/files/library/sp1800/es-idam-nist-sp1800-2a-draft.pdf. [Accessed: 14-Jun-2016].
[109] “NISP SP 1800-2b - Identity and Access Management, Approach, Architecture, and Security Characteristics.” [Online]. Available: https://nccoe.nist.gov/sites/default/files/library/sp1800/es-idam-nist-sp1800-2b-draft.pdf. [Accessed: 14-Jun-2016].
[110] “NISP SP 1800-2c - Identity and Access Management, How to Guides.” [Online]. Available: https://nccoe.nist.gov/sites/default/files/library/sp1800/es-idam-nist-sp1800-2c-draft.pdf. [Accessed: 14-Jun-2016].
[111] “OMB E-Authentication Guidance for Federal Agencies.” .
[112] “Prism Header - martin.cc.” [Online]. Available: http://www.martin.cc/linux/prism. [Accessed: 20-Oct-2016].
[113] “Radiotap - radiotap.org.” [Online]. Available: http://www.radiotap.org/. [Accessed: 20-Oct-2016].
[114] “RFC 494 -2007 Internet Security Glossary,Version 2.” [Online]. Available: http://www.rfc-base.org/txt/rfc-4949.txt. [Accessed: 06-Oct-2016].
[115] “There Is No Privacy On The Internet Of Things.” [Online]. Available: http://www.forbes.com/sites/danielnewman/2014/08/20/there-is-no-privacy-on-the-internet-of-things/#69984ab76b4e. [Accessed: 30-Oct-2016].
[116] “ThreatMetrix - The Digital Identity Company,” *ThreatMetrix*. [Online]. Available: https://www.threatmetrix.com/. [Accessed: 16-Aug-2016].
[117] “Unified Authentication Framework Architectural Overview - FIDO Organization.” .

APPENDIX

LINKS TO DATA SETS

The links below are online copies of the data sets used in the research found in Chapter Two of this paper. Every effort was made to associate the data sets with the various papers. In some cases, the data set was used by two or more sets of research teams.

CRAWDAD (Community Resource for Archiving Wireless Data at Dartmouth) – has a collection of data sets and tools. I am providing the datasets that are relevant.

http://crawdad.org/init/factory/20160613/ no paper but interesting dataset on 802.11 signals in a noisy environment.

http://crawdad.org/gatech/fingerprinting/20140609/ - [81] and [82]

http://crawdad.org/sapienza/probe-requests/20130910/ - no paper associated but a set of probe response request frames.

http://crawdad.org/uw/sigcomm2004/20061017/ - [19], [46], [53], and [66]

https://webtransparency.cs.princeton.edu/webcensus/ - this site has multiple data sets that can be used in a variety of research papers primary emphasis is on websites and tracking across the internet.

https://fingerbank.inverse.ca/api/v1/download?key=dfe76f39e03ac8987e0898c038596e3fb417cda8 – no paper