Attestation Project Thesis

Jonathan Yang

Advisors: Deborah Estrin, Jeff Burke
CS194 Fall 2007 – Winter 2008
University of California Los Angeles

I. Introduction

The Center for Embedded Networked Sensing (CENS) at UCLA has deployed a Wi-Fi mesh network around the UCLA campus as well as the Los Angeles State Historic Park (LASHP) near downtown Los Angeles. The network provides a research framework for participatory sensing, also known as urban sensing. Urban sensing utilizes software and network technology to enable mobile devices to act as credible sensors in environments.

Applications of urban sensing can be used in development of citywide cultural, social and personal applications. A campaign model has been created to integrate the many participatory sensing applications. There are four roles to be played in such a model. The gatherers consist of the mobile devices that take samples of sensor data. The initiator or campaign manager ensures proper execution. The campaign auditor oversees the general campaign. Finally, the analyst is the one who subscribes to the campaign in order to study the data that have been gathered [2].

One example application of urban sensing is a sound level mapping campaign. Wi-Fi enabled cell phones would gather data samples at arbitrary times and upload them to a central server. The data would contain the decibel level of the sample as well as the location and time of the sample. If such a sound map were created in an environment like Los Angeles, it would be helpful towards urban planners and developers. Furthermore, one could monitor the quality of life in the city with such a sound map [2].

When gathering mobile data, the location and time are equally important to data credibility as the gatherer's identity. The more credible the space-time context is, the more useful the data becomes for the purposes they serve. To establish such credibility, the network supporting the urban sensing project is able to attest to the context of the data by tagging data packets with network verified location and time [1]. The goal is to implement the network ability to attest to the location of the sensor data.

II. Attestation Abstract

Attestation is the name of the network ability to attest to the location of sensor data. The mobile devices used are smart phones with GPS functionality. When the phone gathers sample data, it uses GPS to determine the location where the sample was taken and include it into the data of the packet. The network attests to the location where sample was taken by observing the access point, to which the phone connects for internet access, and verifies that the GPS location recorded in the packet is within a certain radius from the known GPS location of the access point. Therefore, the location measured by the network is only as accurate as the wireless reach of each individual access point. The packets that have been attested by the network will be tagged with a flag, certifying that the data location has been verified.

The main purpose of Attestation is to increase credibility of data samples. The primary goal is to prevent technical glitches with GPS calculations on the mobile devices, as opposed to preventing hackers from trying to maliciously alter the GPS data. Participatory sensing operates on the idea that each gatherer willingly participates in a campaign. Therefore, the possibility of users hacking the network data can be safely assumed to be less likely. However, stronger measures to prevent hacking may be implemented in the future.

The network ability to measure the location of mobile device has already been implemented in the past. A common method is to measure the signal strength between the access point and mobile device. This provides an approximate distance to where the device is located in relationship with the access point. If there are multiple access points available, then the network can triangulate the location of the device. As a matter of fact, this technology is already being used ubiquitously in Apple Inc.'s iPhone and iPod Touch. By using network triangulation,

Apple's mobile devices are able to calculate an estimated GPS location by using existing network infrastructure.

However, Attestation is completely different from any past implementations. The goal of Attestation is not to provide an independent measurement of a device's location, but to attest to the location that is initially measured through GPS. Therefore, Attestation is focused more towards verifying the given location of a data sample as opposed to achieving more accurate measurements through network triangulation. The utilization of Attestation is very specific to the area of urban sensing, and therefore, unique in implementation.

III. Installing Nagios

A. Motivation

At the beginning of the fall quarter in 2007, much of the underlying network infrastructure for a small controlled urban sensing environment had been established. Two wireless mesh networks had been deployed on the UCLA campus, one in the north campus Sculpture Garden and one in the south campus Court of Math and Sciences. The mesh networks would provide a test bed for the entire urban research project. However, even though the hardware for the network was operational, many software components for the network still had to be implemented to support urban sensing. Most of the software is still in development today by CENS. Such software includes Campaignr, an application that runs on the mobile devices to collect sensor data and sends it to the campaign database. Another major software in development is Perff, the network database that manages the information collected by each campaign. Almost all of the features required for urban sensing are supported through software running on the network. Attestation is also one of the many features that require software implementation.

The implementation of Attestation involves intercepting packets on the network and analyzing the packets for information used to attest the location the sensor data. Therefore, Attestation is tested by generating traffic on the network and verifying that the packets are intercepted correctly. However, if packets are not intercepted, then the problem could be caused by poorly written software or a network malfunction. To prevent the latter case, CENS found it necessary to implement a network status monitoring system. Such a system would allow constant status updates for every network device, and is imperative to set up prior to conducting any testing on the network.

B. Virtualization

Out of the many network-monitoring software available, Nagios was most appropriate for the CENS network. Nagios is an open source network-monitoring program that offers many configuration options to accommodate any network. It is developed for a Linux environment, which is very appropriate for our network, since many of the servers on the network run a distribution of Linux.

A robust network monitoring system entails installing Nagios on separate machines independent of any existing network infrastructure. Therefore, if one machine had a failure, the other ones would still be able to log the network failure. However, CENS did not have the resources to provide dedicated machines to monitor the network. Therefore, Nagios was installed on the network's namespace server. This meant that if the server were to fail for any reason, the Nagios would stop working as well. Consequently, the decision to install Nagios on a virtualized operating system was made. Therefore, when CENS decides to add dedicated computers to monitor the network, they can easily copy the virtualized image from the namespace server over to the new machines. As a result, installing and configuring Nagios can be bypassed on any new machine in the future.

VMware Server is free software that provides operating system virtualization in Windows and Linux. Because Nagios requires a web browser for the user interface, an operating system with a GUI was necessary, and the desktop edition of Ubuntu was chosen to be virtualized. In addition, since the namespace server's operating system did not have a GUI, VMware Console had to be installed on another computer to remotely connect and use the GUI in the virtualized Ubuntu.

By default, a virtualized operating system uses a Network Address Translation (NAT) network configuration. This resulted in an unroutable IP address given to the virtualized copy of Ubuntu because the IP was being shared with the host. In order for users to connect to the Nagios web interface, the IP address for the host of Nagios must be known. Therefore, the IP must not only be routable, but ideally static. In order to gain a routable address, VMware server was configured to use a bridged network setup. This allowed the virtualized Ubuntu to attain an IP directly from the DHCP server. In addition, the virtualized Ubuntu was configured to obtain a static IP, which would provide users with a static address to connect to the Nagios web interface.

C. Configuring Nagios

Once the setup of Ubuntu was complete, Nagios was installed and configured. The advantage of Nagios being open source is that the application is highly configurable to fit any network setup. Each device on the network is an object written in the configuration file, and every network service for each device must also be individually created through the configuration files. After configuration for Nagios was complete, all devices on the UCLA campus network were monitored. Network services being monitored on most of the devices included Ping, HTTP, HTTS, NTP, SSH, and Vlan. Important devices such as the gateway routers would have all of these service checks. However, some of the smaller devices such at the Wireless Lan Controller do not support all of the services, so only services relevant to the device are monitored.

IV. Implementing Attestation

A. Approach

Although Attestation can be implemented by more than one method, CENS wanted an approach that did not directly involve other existing software. Therefore, even though implementing Attestation might be possible with Campaignr, CENS wanted to use another method. The primary reason for developing Attestation independently from other applications was because Attestation was in an experimental stage of development while Campaignr had already been developed for some time. If Campaignr were to support Attestation, then the complexity of the application would increase without a guarantee that such implementation for Attestation would work successfully.

Campaignr is an application written by CENS and runs on the Symbian Operating System on the Nokia N80 mobile smart phones. When the smart phones gather sensor data, Campaignr processes the data and sends the data over the network to Perff, the sensor database. In order to attest to a packet's location of origin, Attestation compares the GPS location recorded in the packet's data with the known GPS location of the access point (AP). The location is considered verified as long as it falls within a specific radius from the AP. To work independently from other applications, Attestation must intercept packets in the middle of the network as they travel from the mobile device to Perff.

B. Obtaining the Access Point MAC Addresses

The GPS location is known for each AP and stored in a database. Therefore, the verification of the packet origin can be performed once the AP for the mobile device is known. In order to differentiate between the many APs, a GPS location is associated to each access point's MAC address in the database, as opposed to associating the location to the AP's IP

address. As the IP address of an AP can easily change, the MAC address of the AP is static and theoretically unique. Therefore, the database will only have to update an entry if the location of an AP changes, which should happen very rarely.

Due to link layer protocol, MAC addresses between two network nodes are only known during the transmission of data between the two nodes. Once the transmission is complete, the neither node's MAC addresses are stored within the packet. This suggests that as a packet is sent from the smart phone to the database, the MAC address of the AP is known at some point in time, but is lost before the packet reaches the database. The goal is to determine which node on the network is able to see the MAC address of the AP.

In a simple network, the receiver to whom the AP sends the data packet is the node that knows the MAC address of the AP. However, the CENS network uses Cisco network devices, which operate with proprietary protocols. Therefore, it is not obvious to which device on the network is able to see the MAC address of the AP. For a layout of the network, refer to Appendix A. As previously mentioned, there are mesh networks located in the north and south side of the UCLA campus. Both networks have identical, and therefore, all testing was performed on the south campus site.

At each site, there is an AP that has a wired connection to the router. These wired APs are called the Root APs of the network. The other APs in the same mesh network communicate with the Root AP wirelessly and send all of their traffic through the Root AP to reach the router. On the network map, the Root AP is connected to the Wireless Lan Controller (WLC), which is connected to the gateway router. The WLC is a Cisco device that manages the network APs for the router. However, after inspecting the physical connections between the Root AP, WLC, and router, the connections did not exactly follow the ones shown on the network map, but resembled a setup shown by the diagram in Appendix B. The traffic from the root AP is sent to the router first, which then passes it to the WLC for packet processing. From the WLC, the traffic is returned back to the router where it is sent towards the database.

Since Cisco uses a proprietary protocol for communication between the devices, it was unclear to when the MAC address of the AP was lost. Therefore, a tool like Wireshark was used to analyze packet information. Wireshark is a packet analyzing application that sniffs packets and displays the contents within the packets. The router was configured to mirror all traffic from the router incoming from the Root AP and the WLC to another computer running Wireshark. By using a laptop, test traffic was generated over the network while Wireshark sniffed the packets. The results were that the packets incoming into the router from the Root AP contained the AP MAC address while the traffic incoming from the WLC did not.

Initially, the laptop was generating traffic while being directly connected to the root AP. Because all the other APs forward their traffic through the Root AP to reach the router, it was uncertain if traffic generated on another AP would show the MAC address of the original AP or Root AP once it reached the router. Therefore, test traffic was also generated with the laptop connected to a different AP. The packet information showed that indeed, the original AP MAC address was stored inside the packet. Therefore, the conclusion was made that Attestation must intercept the packets incoming into the router from the Root AP because that was the only location in the network where the AP MAC address was known.

C. Implementing Herbivore

With the knowledge of where to extract the AP MAC address on the network, an implementation for Attestation was finally possible. When Campaignr sends HTTP packets to the Perff database, the information about the packets known on both ends are the source IP and port, destination IP and port, packet checksum, and packet timestamp. Clearly, what is not known on both ends is the MAC address of the AP that the mobile device used. Therefore, a program is needed to capture packets coming from the Root AP to the router. This program, named Herbivore, logs information about packets known to Perff as well as the AP MAC

address. Therefore, when Perff receives a sensor data packet, it can look for an entry in the Herbivore database that matches the packet source IP/port, destination IP/port, checksum, and timestamp. The timestamp will make the packet unique from others, even if there are multiple packets being sent from the same mobile device to Perff. Once a match has been found, the AP MAC address can be easily obtained from the database as well.

Because Herbivore must constantly be able to intercept packets, a dedicated machine was used for this purpose. The traffic from the Root AP to the router was mirrored to a dedicated Network Interface Card in the machine. Herbivore is written in the C language, and in order to capture packets, the pcap library is used. The pcap library is the same library that Wireshark and many other packet capturing applications use for all sniffing functionality.

After generating and examining more test HTTP packets, the complexity of the packets was realized. The packet has many layers of encapsulation. From the outer to innermost layer, the packet encapsulation order is Ethernet II, IP, UDP, LWAPP, IEEE 802.11, Logical-Link Control, IP, and finally TCP. The contents of a test packet can be view in Appendix C. The outer IP layer is used only for communication between the AP and the WLC. The LWAPP layer is Cisco's proprietary protocol for communication between its devices. In the IEEE 802.11 layer is the AP MAC address which needs to be extracted. The inner IP layer is where the source and destination IP are located, which also need to be extracted. Lastly, the source and destination ports as well as the checksum are located in the TCP layer. In order to obtain the needed fields for database information, Herbivore must parse the packet starting from the outermost layer. To achieve this, structures of the protocol headers must be defined. When the packet is read, the data is typecasted to the appropriate header field. The next layer is reached by accessing the data payload of the current layer.

Because TCP protocol does not have timestamps, another method must be used to time-sync the packets on Herbivore and Perff. Firstly, the machines running Herbivore and Perff must run the Network Time Protocol to synchronize the time between the machines. Both Herbivore and Perff must log the receiving time of the sensor packets. The difference between the two receiving times for the same packet should be on the magnitude of milliseconds. Therefore, when Perff looks for a matching entry in the Herbivore database, it can find an entry with a timestamp that may differ by only a few milliseconds. If multiple sensor packets were sent during this time frame, there may be multiple matches in the database. However, it is extremely unlikely for the mobile device to change locations during such a short time span. Therefore, the verification of location can be made towards all sensor packets within such a time span.

Another method is to use Campaignr to write the time when the sensor data was gathered into the packet. Therefore, both Herbivore and Perff would grab the same timestamp written by Campaignr in the packet's data payload. However, Campaignr is still under development and it is uncertain if such a feature has yet been implemented.

Currently, the development of Herbivore is still in progress. Several protocol header structures have been defined such as TCP, UDP, and IP. However, the other layers must also be defined in order to parse the packets correctly. Packet capturing functionality has already been implemented and Herbivore is able to print simple fields like the length of the packet in bytes. Although my research with Herbivore and Attestation will conclude with the end of the Winter 2008 quarter, there will be other undergraduate students who will continue this project in the coming quarters. Therefore, Attestation will continue to be implemented until it is completed.

V. Alternative methods

As mentioned before, Attestation can be implemented with multiple methods. Another choice would be to make Campaignr include the AP MAC address into the packet data along with the other sensor data. Since the mobile device must send the packet through the AP, it must know

the MAC address of the AP as well via link layer protocol. If that information is not easily accessible, a single Address Resolution Protocol request sent by the mobile device can also determine the MAC address of the AP. With the MAC address written into the packet data field, the calculation to verify the location of the sensor reading can be done anywhere on the network.

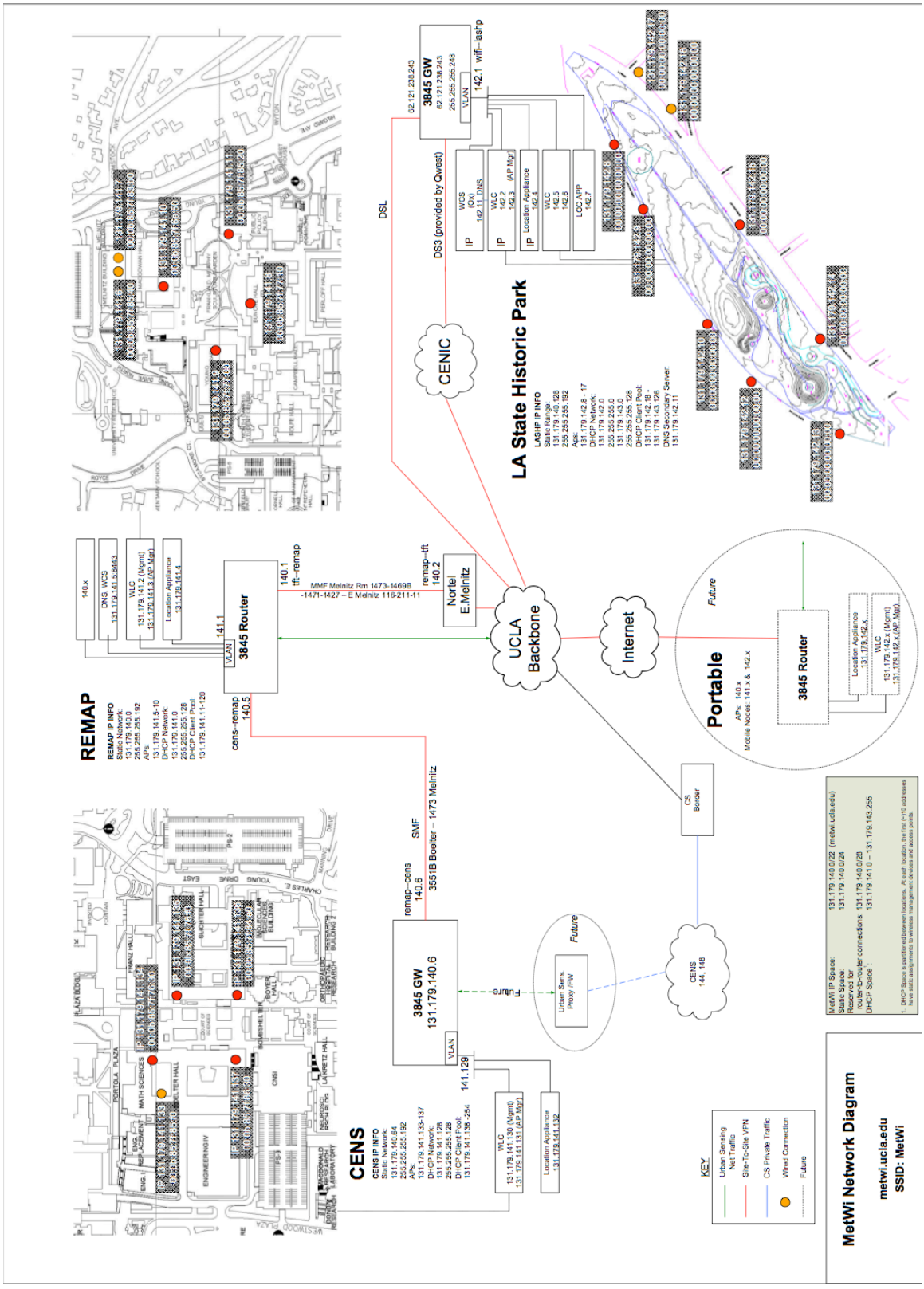
However, there was one main reason why this approach is not taken. The mobile devices CENS uses are Nokia N80 smart phones that run the Symbian operating system. From the experience of the researches that have been implementing Campaignr, programming for Symbian is very difficult and has a steep learning curve. Therefore, adding Attestation functionality through Campaignr would be very challenging. Additionally, it has the possibility of breaking existing features in Campaignr, which is a risk the Campaignr team is not willing to take. Therefore, although this approach can theoretically be implemented, it is not because of the technically challenges that entail.

VI. Conclusion

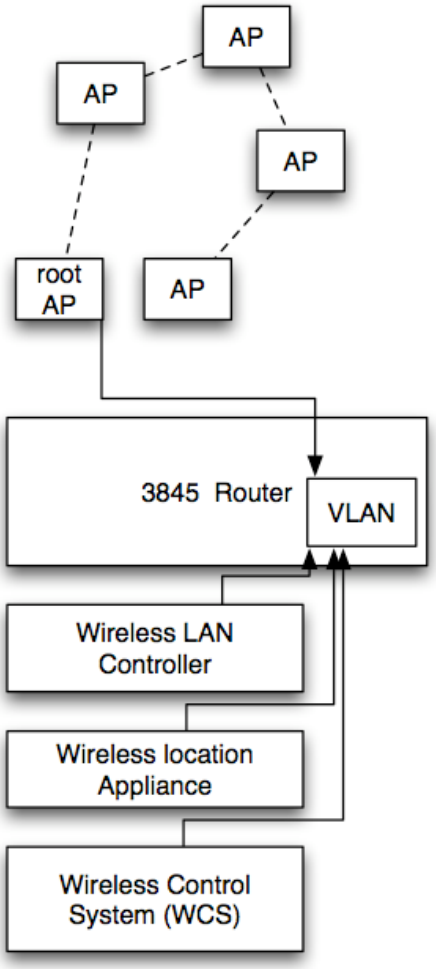
This paper has described the concept of urban sensing and the importance of location credibility with data samples. To attain such credibility, Attestation is created to implement a network ability to attest to data sample's recorded location. The first quarter of research was dedicated to installing and configuring Nagios for the network. Such network monitoring application was necessary before sending any test traffic over the network because CENS needed to know if any failures in tests were caused by network hardware or services. The second quarter was dedicated to implementing Attestation. The first obstacle was to pinpoint which network device was able to see the MAC address of the access points within the packet. After performing tests, the gateway router was confirmed to be the device. Consequently, the design architecture of Attestation began and Herbivore was created. This confirmed that Attestation was indeed possible to implement, which was unclear at the beginning of the research. Finally, the necessities and steps to extract the required packet header fields, including the AP MAC address, were outlined so that future researchers working on Attestation may continue the work smoothly. Overall, the research conducted over the past two quarters has been successful.

VII. Appendix

A. Network Map



B. Router Connection Diagram



This diagram shows how the network components are connected. Mobile devices that connect to an AP generate packets with sensor data samples. The AP sends to packets through the root AP to the 3845 Router. The router then sends the packet to the Wireless LAN Controller, which processes the packets before sending them back to the router.

C. HTTP Packet Contents

Frame 8 (1446 bytes on wire, 1446 bytes captured)
Ethernet II, Src: Airespac_7f:76:30 (00:0b:85:7f:76:30), Dst: Cisco_66:43:e3 (00:19:e7:66:43:e3)
Internet Protocol, Src: 131.179.141.134 (131.179.141.134), Dst: 131.179.141.131 (131.179.141.131)
User Datagram Protocol, Src Port: 63331 (63331), Dst Port: 12222 (12222)
LWAPP Encapsulated Packet
IEEE 802.11 Data, Flags: ....R...
  Type/Subtype: Data (0x20)
  Frame Control: 0x0809 (Normal)
    Version: 1
    Type: Data frame (2)
    Subtype: 0
    Flags: 0x8
      DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
  Duration: 12288
  Destination address: Airespac_7f:76:30 (00:0b:85:7f:76:30)
  Source address: Apple_c4:92:d2 (00:1b:63:c4:92:d2)
  BSS Id: Cisco_c0:61:f0 (00:1a:6d:c0:61:f0)
  Fragment number: 11
  Sequence number: 777
Logical-Link Control
Internet Protocol, Src: 131.179.141.176 (131.179.141.176), Dst: 209.85.173.147 (209.85.173.147)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..0. = ECN-Capable Transport (ECT): 0
    ....0 = ECN-CE: 0
  Total Length: 1366
  Identification: 0x1447 (5191)
  Flags: 0x04 (Don't Fragment)
    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (0x06)
  Header checksum: 0x910e [correct]
    [Good: True]
    [Bad : False]
  Source: 131.179.141.176 (131.179.141.176)
  Destination: 209.85.173.147 (209.85.173.147)
Transmission Control Protocol, Src Port: 53169 (53169), Dst Port: http (80), Seq: 1, Ack: 1, Len: 1314
  Source port: 53169 (53169)
  Destination port: http (80)
  Sequence number: 1   (relative sequence number)
  [Next sequence number: 1315   (relative sequence number)]
  Acknowledgement number: 1   (relative ack number)
  Header length: 32 bytes
  Flags: 0x18 (PSH, ACK)
  Window size: 65535
  Checksum: 0xcdf9 [correct]
    [Good Checksum: True]
    [Bad Checksum: False]
  Options: (12 bytes)
    NOP
    NOP
    Timestamps: TSval 498475673, TSecr 4211957838
Hypertext Transfer Protocol

This shows the contents of an HTTP packet generated by a laptop. The items in bold are the packet layers that contain fields useful towards implementing Attestation. The items in bold and italicized are the fields that Herbivore must extract. Items that are tabbed represent the content in the appropriate layer. Only layers with relevant information were expanded.

VIII. Bibliography

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava. "Participatory sensing." World Sensor Web Workshop, ACM Sensys 2006. Boulder, Colorado, 2006.

[2] Vidyut Samanta, Jason Ryder, Sona Chaudhuri, Jeff Burke, Deborah Estrin, Fabian Wagmister. "UCLA/Cisco Metropolitan Wi-Fi Research Network. "Computer Science Department, UCLA. Los Angeles, California.

IX. Acknowledgements

Many thanks for the help and guidance from the following individuals:

Deborah Estrin

Jeff Burke

Jason Ryder

Vidyut Samanta

Ryan Dorn

Richard Guy

Derek Kulinski