Wi-Fi tracking : what about privacy

Levent Demir

▶ **To cite this version:**

Levent Demir. Wi-Fi tracking : what about privacy. Mobile Computing. 2013. hal-00859013

HAL Id: hal-00859013
https://inria.hal.science/hal-00859013

Submitted on 6 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Security, Cryptology and Coding of Information systems
Master of Science in Informatics at Grenoble

Wi-fi tracking : what about privacy ?

Levent DEMIR

03/09/2013

Research project performed at INRIA
Under the supervision of:

Mathieu CUNCHE, internship supervisor
Jean-Louis ROCH, university tutor

September 2013

Contents

Acknowledgement

I would like to thank my internship supervisor at Inria, Mathieu CUNCHE, for his assistance regarding my internship in the research center. Although Cedric LAURADOUX, for his availability to help and advice me.

And of course, I thank my internship tutor, Jean-Louis ROCH, for his proposals and assistance regarding my project. Then, to all the teachers of the Master of Computer Security, Cryptology and Information Coding from which I learned in a deep level the essentials of cryptography, systems and network security which aid me accomplish the task assigned by the internship.

Finally, special thanks to all the people working in PRIVATICS Team which helped me throughout the context of my final master internship, and to CLAUDE CASTELLUCIA who gave me the opportunity to perform my internship and apply what I've learned from my master classes in Computer Security, Cryptology and Information Coding throughout the academic year.

Presentation of INRIA and PRIVATICS

Created in 1992, the Inria Grenoble - Rhone-Alpes centre was established then developed in partnership with the clusters and institutions of Grenoble and Lyon. The center currently has 34 research teams, including 11 located in Lyon.Inria Grenoble is a major player in research and innovation in computational sciences in the Rhone-Alpes region. The research center contributes especially to the fields of embedded software and imagery in Grenoble and to information technologies applied to life sciences in Lyon.

Privacy Models, Architectures and Tools for the Information Society (PRIVATICS) is a research team specialized about privacy issues. The goals of the Privatics team is to study the new privacy threats introduced by the information society and design privacy-preserving solutions to prevent or at least mitigate them. The project follows a multidisciplinary approach. It focuses on technical and scientific problems, but also considers the economical, legal and social aspects of privacy. The research themes are to understand and formalize privacy, to build privacy preserving systems: Privacy-by-Design, Privacy and Transparency Enhancing Technologies.

The project I am working is about the privacy concerning the Wi-Fi. We want to know how tracking companies manage the monitored data. The team is although working about the privacy concerning the smartphones' applications of Android and iOS operating system. The goal is to understand which personal data is sent without the agreement of users.

Abstract

Tracking individuals is a major issue today. Several methods exist, but they are limited you to user interaction. A new way of tracking is based on Wi-Fi which does not need any user cooperation. However this method involves new privacy threats.

This paper verify if the existing solutions to protect the privacy are secure enough. The tracking system is based on Wi-Fi sensors installed in different locations to monitor and record packets sent by Wi-Fi-enabled devices. The identification process is possible thanks to the MAC address contained in each packet. However this unique identifier is bound to the user and has to be securely stored. Tracking individuals has plenty of applications. Studying the habit of customers in shop center is a service offered by several new specialized companies. We analyze which kind of protection are used, where and how data are stored. One commonly used protection is the one-way hash function. Studying the MAC[1] addresses' distribution leads us to develop an attack. Finally, a test platform is built to validate or not the solutions or think to new ones.

I. Introduction

The usage of mobile phones equipped with a Wi-Fi interface increases considerably. Wi-Fi-based-tracking-system have followed the same trend. The principle is based on nodes capturing packets over the air and recording some information about a discovered device such its signal strength or MAC address. Those data allow a tracking system to track individuals and to compute statistics over a group of individuals. The two main issues about this process is that users are unaware of all those equipments. Secondly, the privacy of tracked individuals could be threatened because of weaknesses in the anonymization process.

In the literature, a lot of works are about wireless localization [24] or authentication for local area networks [8]. Other works are based on wireless beacons, studying the link between wireless devices' owner thanks to the study of the broadcasted ssid [10].

Our work was based firstly on an analyze of packets send by Wi-Fi enabled devices. We notice that a lot of frames could be considered as identifying frames, which contain a unique identifier, the Media Access Control (MAC) address. The problem was to understand how those informations were stored in database. We discover that a commonly used protection method is to use a one-way hash function. Next, we develop two attacks to de-anonymize the protected data. Our attack model in this paper is to study the consequence of a database leak containing all the protected MAC addresses, and more particularly the deanonymization of this MAC addresses and the potential revelation of the owner's identity.

The structure of the paper is as follows. Information about Wi-Fi technology and communications is provided in §2. We describe the principle, applications and privacy issues of Wi-Fi based tracking systems in §3. In §4, de-anonymization methods of the Wi-Fi based tracking database are described followed by the experiment results in §5. Two counter-measures and a test platform are presented in §6. §7 concludes this paper.

II. The Wi-Fi technology

The Wi-Fi technology has been developed to allow different devices to communicate without any wired connection. During a communication between a Access Point (AP) and a station (STA), several frames are exchanged. Those frames are divided into three types (Management,data and control) following the MAC frame format shown in figure 1. Each frame is composed of :

[1]Media Access Control

- A MAC header, which contains frame control, duration, address, optional sequence control information and other optional fields.
- A variable-length frame body with some specific information to the frame type and subtype.
- A FCS, which contains an IEEE 32-bits CRC.

The first field Frame Control (FC) gathered several information, the Figure 2 shows the eleven subfields. The Type and Subtype subfields represent the purpose of the frame, it could be control, data or management. For example the probe request corresponds to the 0x04 value, the probe response is the 0x05. All the details of different types are available in the appendix A. The most useful frames for our research are frames from the Management Type (Probe request/response, beacons) and from Data Type (Data, null function and Qos Data) because they are sent periodically and are plenty.

The next subfield is the Power Management, it indicates the power management mode of a STA. The second field is the Duration/ID, the next four fields are addresses, for the basic service set identifier (BSSID), the source address (SA), the destination address (DA), the transmitting STA address (TA), and receiving STA address (RA), some of those fields are reserved to certain frames.

The Sequence Control Fields is composed of the fragment number and the sequence number.

The length of the frame body depends on the frames type.

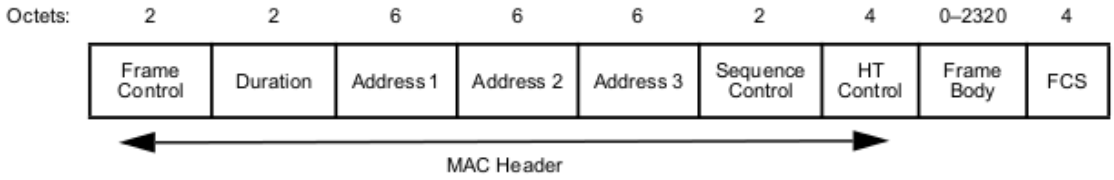

Figure 1: *Mac frame format*

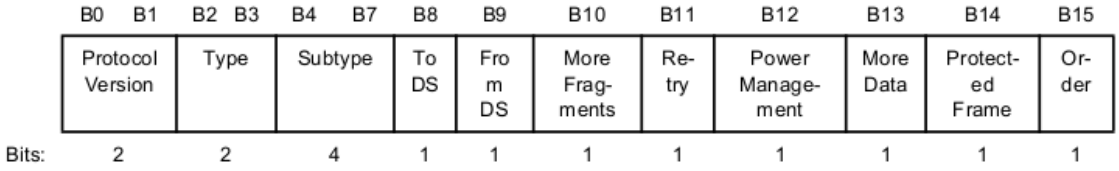

Figure 2: *Frame control field*

All frames contain the MAC address which is a unique identifier for all network interface. Every smartphone or tablet has at least two MAC addresses, one for the wireless card and one for the bluetooth card, computers have also a third one for the physical interface. A MAC is composed of 6 bytes, the first three are reserved and only delivered by the IEEE. This prefix is called OUI "Organization Unique Identifier" or "companyid". The range of all MAC addresses represents 2^{48} addresses.

The question of energy consumption is crucial for embedded systems as mobiles phones or other mobile devices. The most energy-friendly solutions are adopted, one is present in the Wi-Fi

service discovery, a second is set during the communication between a STA and an AP. and some special frames.

II.1 Wi-Fi service discovery

The Wi-Fi is based on many spectrum bands, the main used is the 2.4 Ghz band. It can carry a maximum of three non-overlapping channels. There are fourteen channels defined in the 802.11 protocol (Wi-Fi)for the 2.4 Ghz ISM bands. The channels are not all allowed in all countries, some are used in north America (11 channels), for Europe 13 channels are available. An AP can be in any of those channels that is why during the AP research for a device, more than one broadcast are sent. This process is called Wi-Fi service discovery which is divided in two modes [15] :

1. *Passive service discovery* : the AP is sending periodically beacons which are broadcasted, the stations are listening passively and switching through the set of possible channels.
2. *Active service discovery* : stations are sending probe requests, they try to find known AP by probing each channel. It also could be a broadcast, when the device sends a beacon, all the AP around are responding with a probe response.

The first mode could be time-consuming due to the amount of channels. The device is in listening mode during a long period, this is more costly than emitting during a brief time. That is why the active mode is less energy-consuming, it is privileged in practice. Devices are sending probe requests only when necessary.

II.2 Energy saving mode

An other way to preserve the battery life of mobile devices is the use of the power management frames. Smartphones are going sometimes offline for two reasons. The first one is because of the power save mode. The system is turning off the radio to preserve the battery life time. The second reason is to analyze the around AP to look for a better wireless signal or when the station has hit its roaming threshold. During this offline time, the STA does not receive any frame, for this reason the power management bit of the frame control field is used to inform the access point of the state of the station. For example, when the STA sets the bit to 0, it means that "STA will stay up" and "STA will go to sleep" for 1. This process allows the STA to save energy by telling the AP to buffer the data and send when available. This frame can also be used to track a device as any other Wi-Fi frame, it contains the MAC address of the source.

II.3 Frequency of probe requests

II.3.1 Description of frames

The 802.11 standart has defined different frames type used by stations for communicating, managing and controlling the wireless link. Some of those frames are considered as identifying frames, they are useful for tracking thanks to the MAC address present in each packet. In our research, we are focusing on management frames. The first managing frames used to track individuals is the probe request frame. This frame is sent by the device during an active research to discover around AP. Another interesting frame is the null data frames. The purpose of this experiment is to understand the probing frequency of a small yet representative set of mobile devices.

II.3.2 Experiment protocol

The study is about Android/Apple phones and tablets. Two modes are available and have different behaviors. The first mode called active mode is when the screen is on and the second mode called idle mode is when the screen is off. Moreover, two more case are studied. First, when the device is connected to an AP and second when the device is not connected and is searching AP. In total, we have four configurations for each device. Before starting the monitoring, the wireless has to be on at least 30 minutes on, indeed, the behavior at the start up of the communication is different from when they are established. The experiment protocol is described below and the results in Table 1.

1. Active mode
    - The experiment lasts one hour. No preferred networks saved (forget around networks). Screen is on. Start the record before waking the phone up.
    - The experiment lasts one hour. Preferred networks saved. Screen is on. Start the record before waking the phone up.
2. Idle mode
    - The experiment lasts one hour. No preferred networks saved (forget around networks). Screen is off.
    - The experiment lasts one hour. Preferred networks saved (forget around networks). Screen is off.
3. Special tests
    - What happened when the device is waked up (switch from idle mode to active mode)
    - What happened when the user go to the wireless parameters, is the device automatically looking for networks ?

II.3.3 Results

The results are shown in Table 1, the frequency of probe requests is higher for the active mode and without any connection. That is explained by the fact that the device is actively looking for an AP. In this case the average interval of time between probes is 10s. Then, this time can be multiplied by 2 each cycle until a fixed value.
When the device is in idle mode, the frequency is not regular. It could be every 10s but after the time increases rapidly to 500s or 1000s. For example for iPhone 4S, in 83 minutes there are only 6 probe requests. Moreover one can notice an interesting fact for iPhone with iOS version above 6.0, when the device goes to idle mode, the connection is deassociated, then after 300 s deauthenticated. The connection is recovered when the device comes back to active mode.
The privacy of users was compromised because of the diffusion of the known SSID, the results show that Apple devices are sending more easily the known networks SSID compare to Android devices.

Finally, the connection to an AP is useful for tracking because a lot of special frames as Data or Null frames are sent. This means that if a user is using his device during the movement, each second a frame which allows the recognition thanks to the sent MAC address.

Devices		Active mode		Idle Mode	
		Without connection	With connection	Without connection	With connection
Apple	iPhone 4S	Every 10s until 160 after every 50s	A few probe request- Several QoS Data and Null frame	10-10-50-50-100-100	A few probe request- Several QoS Data and Null frame
	IPAD 2	Every 10s until 120s then 30-60-120-240-500-500-500	Probe request every 10s - Null function depending on Safari usage	10-10-1000s	idle 2
	iPhone 5	20-20-30-60-120-240	Null function depending Internet usage	0-20-200-600	5-10-10-20-30-60-10
Android	Samsung Galaxy S3	0-10-20-35-65-130 then every 130 s until 30 minutes	10-50-100-200 periodically	Every 200s	Null frame every 60s
	Galaxy Nexus 4	Every 20s	active 2	idle 1	idle 2

Table 1: *Results of probe requests - The value indicates the interval between two probes -*

II.4 Monitoring Wi-Fi communications

The wireless technology make the life easier for users, however from a security and privacy point of view, several threats appear. Any malicious person can have access to the Wi-Fi connection and can potentially attack the device. To monitor Wi-Fi communications, two tools are required, one named **aircrack-ng suite** has been created at the beginning as a set of tools for auditing wireless networks. This open-source software could be installed in equipment with adapted hardwares with special wireless cards. The drivers also should be correctly installed. A few works [19] are focused on the installation of the aircrack-ng suite in smartphones. That means that the nodes could be easily nomad. The second tool named **tshark** for the command line version is used to analyze the recorded frames with a powerful filtering system. The user can use complex filtering rules for searching some recorded packets. Or one can directly apply rules during the record.

Those tools allow us to work in two modes. The fist mode is the monitoring mode which consists in recording all the diffused packets, this mode is passive and very efficient. It requires no modifications of devices, the action area is the coverage area of the monitoring device. The other mode is the injection mode which allows the user to apply replay attack. The attacker sends packets of a pre-recorded or well-known access points frames in order to simulate a legitimate AP.

This mode could be used to force the communication with devices.

In our case the main used mode is the monitoring mode, we are passively recording all the identifying frames (containing the MAC address of the source and the destination).

III. Physical tracking systems

III.1 Introduction

A lot of companies try to understand their customers behavior. Monitoring a web-based commercial service is easy thanks to the purchase history, google analytics and others tools. However physical shops are not able to have precise statistics about the customer's behavior. They try to count by hand, with sensors or advanced face recognition using digital cameras. A new trend for monitoring exploits wireless communications. Some research have shown [18] that 70 % of all smartphone-originated traffic is coming from the Wi-FI. The potential of tracking is significant thanks to the amount of people having the wireless enabled.

III.2 Principles

The principle of wireless-based monitoring is to install a network of nodes (able to monitor all the devices around in a certain perimeter) and record some useful information as the signal strength and the MAC address in order to locate the devices' owner with triangulation. When the architecture of nodes is carefully deployed, the accuracy could reach the meter. The system is able to track all devices with wireless interface as smartphones, tablets, computers.

III.3 Applications

Several companies have created their business on this technology. The objectives are several, the owner of the system could be interested in counting unique MAC addresses -user- in order to make statistics. One can be interested for the geopositionning of a user or making a map of the movement of the individual.

III.3.1 Physical analytics

The more popular use-case is to use the Wi-Fi technology to make statistics about population traffic and movement. The purpose is to reveal and found a global behavior of customers in shops or crowd in stadium, trade shows ...

The system is based on a network of nodes which cover an area (site). Those nodes are installed on strategic point separate to cover all the monitored area. Then when a device enter into the monitored site, it is immediately found. Some information as device ID and signal strength are sent to the cloud process. The aggregate information from multiple nodes provides accurate positioning of each Wi-Fi device using algorithms. Then thanks to an online interface the user can see all devices represented in different customizable maps.

Two companies' main activity is the physical tracking system.Navizon is specialized in the geopositionning for smart phone users [5]. The principle is illustrated in figure 3. The second company is Euclid [2] specialized in tracking users, it describes itself as something of Google analytics solution of the real world. A new interesting solution has been developed, the concept is to use the existing wireless infrastructure to detect the shoppers.

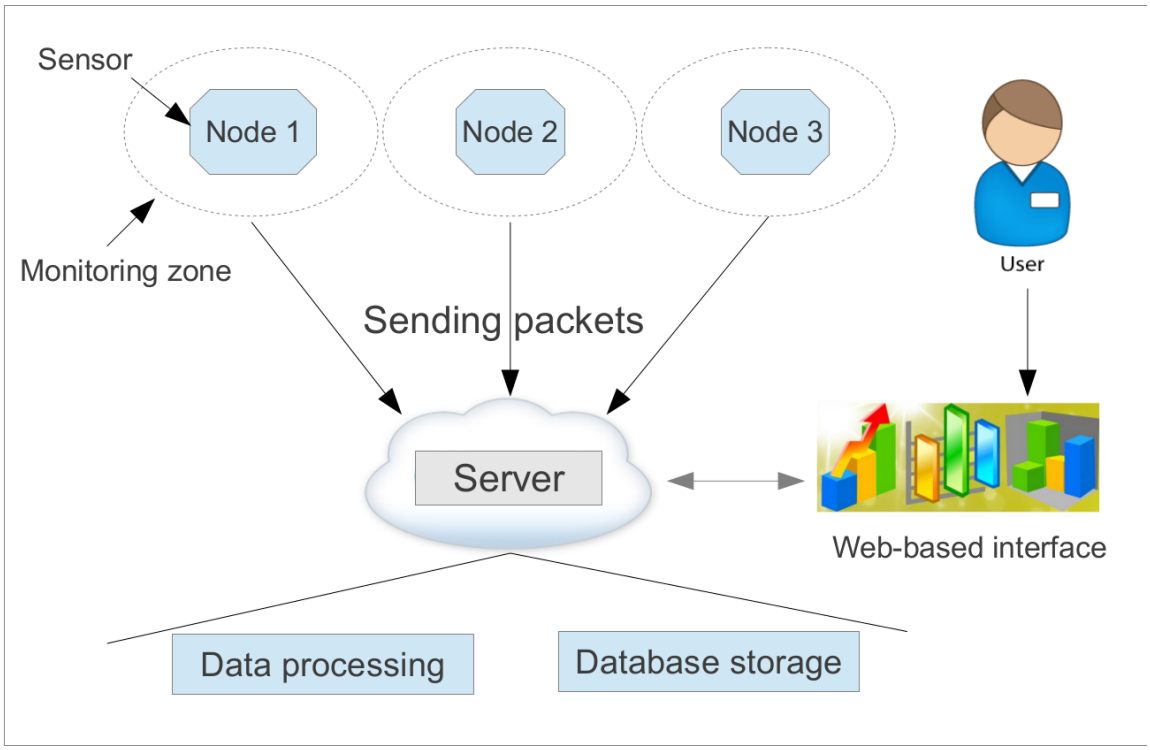

Figure 3: *Overview of Wi-Fi based tracking system*

III.3.2 Traffic control

The Wi-Fi is a good way of tracking individuals for traffic control. For example, a research team has studied [16] the traffic in a single road-side.They observe that in one 12-hour trial on a busy road they have detected 7,000 unique devices. This means that not all devices have been detected because of the speed and the brief time exposed in the coverage area.The ratio is one device for 5 vehicles (based on the statistic of the state DOT : 37,000 vehicles daily counted). However if a 7 nodes mesh is installed in a 2.8 km road, the detected devices is up to 68 %. This method is used to monitor the traffic and although the trajectory of vehicles and individuals. The mean error for all tests is 69 meters. The conclusion is that tracking unmodified smartphones using Wi-Fi monitors is both practical, economical and accurate. Libelium [3] is a company specialized in creating open source sensor platforms, they purpose a motion sensor which can be used to monitor shopping or street activities.

III.3.3 Detecting and locating individuals

Knowing the location of a doctor in an hospital or firefighter deployment in a building is critical in many occasions. Companies may also benefit from the knowledge of the instant employees map to improve logistics or work efficiency. Indoor Wi-Fi is perfectly adapted for this task, the principle is easy to set and economical; the location could be very accurate with an elaborated mesh.

III.4 Privacy issues

Most people are unaware that their Wi-Fi is a potential source of tracking. Each device is linked to a unique identifier (MAC address) and so to an individual. Monitoring and storing this kind of data is a threat for the privacy. Companies deploy several mesh of nodes in different area : individuals could be tracked in a large scale. Risks are higher if those localizations are correlated with other information.

This risk has been taken into consideration by some tracking companies. The main answer to the privacy breaches is cryptographic functions. Let us illustrate their think with some citations taken from the official website :

- Euclid [2] : “ Data is transferred securely (using SSL) and is "hashed" (scrambled into a meaningless string of numbers and letters) before it is stored on Amazon Web Services. Hashed data can not be reverse engineered by a third party to reveal a deviceś MAC address. This means that anyone who gains access to the database directly from Amazon - authorized or unauthorized - will only see long strings of numbers and letters. They would not be able to get any information that could be linked to a back to a particular mobile device owner. ”
- Libelium [3] : “ The anonymous nature of this technique is due to the use of MAC addresses as identifiers. MAC addresses are not associated with any specific user account or mobile phone number not even to any specific vehicle. Additionally, the ”inquiry mode“ (visibility) can be turned off so people have always chosen if their device will or won’t be detectable. ”

Depend on the company, some protections are taken. The chosen solutions are to use a one-way hash function or to consider the MAC address as a non-significant information about users. The following section described if the chosen solutions are secure enough.

IV. De-anonymization of physical tracking database

IV.1 Attack model

Protected data are stored in databases. They contain several informations as the discovered MAC address, the date, the localization. In our research, we are focusing on a model where if there is a leak of the database, is an adversary able to retrieve the original MAC address. We consider the adversary with different levels of risk. In first level the adversary has only access to the database. The second level is to get the system with the database and other data like secret keys if exist. The third level is to have a full access to the system. The main consideration is the first, because in the litterature leak of database is a common situation but taking control of the server is more difficult. Moreover if the adversary has this control, all the incoming information will be in plaintext or less protected, and the data easily stolen.

In addition, the adversary is not always a malicious attacker. Today legal entities like police or intelligence service are asking to have access to this kind of database when necessary. Those aggregate data could although value money. Personal data could be sold to trend analysis companies [6].

IV.2 Anonymization using cryptographic hash function

The purpose of physical tracking system database is to store the recorded MAC addresses and other information in a secure way and by following a few rules. The obtained protected value has to be unique, because if the same individual came again in the monitored area, the API has to recognize him.

To secure the database, companies have chosen cryptographic hash function. A hash function takes a message of any size as input and produces an output called a hash-value or simply hash. Let define two types of hash functions.Unkeyed hash function takes a single input (message) but keyed hash function takes two distinct inputs, a message and a secret key. In our case, we focus on unkeyed version which is used by the most physical tracking companies. Further, we will discuss about the second version in the counter-measure part.

The formal definition [7] is that a hash function is a function h which has the following properties :

1. *compression* - h maps an input x of arbitrary finite bitlength, to an output h(x) of fixed bitlength n.

2. *ease of computation* - given h and an input x, h(x) is easy to compute.

To define more precisely the security of unkeyed hash functions, three properties [7] can be added, for a function h with inputs x, x' and outputs y,y'.

1. *preimage resistance* $\equiv$ *one-way* - for essentially all pre-specified outputs, it is computationally infeasible to find any input which hashes to that output, i.e., to find any preimage x' such that $h(x') = y$ when given any y for which a corresponding input is not known

2. *2nd-preimage resistance* - it is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a 2nd-preimage $x' \neq x$ such that $h(x) = h(x')$.

3. *collision resistance* - it is computationally infeasible to find any two distinct inputs x, x' which hash to the same output, i.e., such that $h(x) = h(x')$.

Based on those properties, several algorithm exist in the literature. One of the first widely used was MD5 [22] algorithm which produces a 128 bit hash value. It was designed by Ron Rivest in 1991 to replace the earlier version MD4. However weaknesses have been founded, MD5 was not collision-resistant [11] [9] [12] [23]. In 1995, the NSA[2] has developed two other hash functions, SHA-0 and SHA-1. They are published by the NIST[3] as a U.S Federal Information Processing Standard. However attacks [25] have weakened the collision resistance of these functions. Those attacks are possible but last a considerable time. In the following part, we will show that some known attacks are very efficient in our case.

IV.3 Attack methods on hash-value

The case of retrieving an original text from the digest is common for the passwords. An adversary gets the /etc/passwd file of a linux file system and wants to recover all the passwords.Some known attacks are used for this case. We describe two of them.

The bruteforce attack consists in trying all the possible value until the correct one is found. The security is based on the length of the key. Suppose the key length is n bits, in the worst case it can be broken in 2^n operations. The limit is that the require resources for such attack is growing exponentially to the size of the key, not linearly. So the larger is the key the best is the protection against brute-force attack.

The principle of the dictionary attack consists in successively trying all the words in an exhaustive list called dictionary. This pre-defined wordlist contain the most used passwords. The

[2]National Security Agency
[3]National Institute of Standards and Technology

optimized version of this attack is the pre-computed dictionary attack. The aim is to achieve to time-space tradeoff by pre-computing a list of hashes of dictionary words, and storing these in a database using the hash as the key. This allow to look up the hash-value almost instantly at any time to find the corresponding password. This attack is particularly effective for big amount of passwords. The advantage is that list of hashes is computed only once and not for each cracking process. However for huge dictionary file, the storage is costly. An other approach is to use rainbow tables [17], which reduce storage requirements at the cost of longer lookup times.

IV.4 Optimized prefix attack

Monitoring identifying frames give us an overview of the most used devices and the distribution of them. We are working within a small domain, a MAC address is composed of 48 bits which offer 2^{48} possible addresses. If we consider that the distribution of addresses is uniform, the brute force attack is adapted to solve the problem. However the MAC address format gives the opportunity to develop a smarter attack, the first 3 bytes are fixed and correspond to the OUI[4].

First we can focus on the OUI distribution. The list is managed by the IEEE registration Authority [1]. The total amount of possible OUI is 2^{24}, the registered amount of OUI is 18189[5]. That means that less than 1 % of OUI are used. A chart of the number of OUI used per constructor is shown Figure 4

The first constructor is CISCO, a world leader in networking. Then todays most smartphones sellers are present. Apple and Samsung are at the second and third place. The chart shows that if we focus on the main constructors, the results will be more fast and efficient.

An other approach is to make our own distribution of most used prefixes. In Lyon, a few thousands of MAC addresses have been recorded.The distribution of those addresses help us to determine how many prefixes are required in order to detect the majority of devices found in this place. We have found that 111 prefixes allow us to find 99.9% of devices, the distribution is shown in figure 5.

To generate the values to be tested, we proceed as follows : first we select a prefix from a list of most frequent OUI. Then, we generate any possible suffixes. This attack is shown in Figure 6.

V. Experiment results

V.1 De-anonymization tools

The need to de-anonymize is similar to password cracking. Two cracking tools are known, John the ripper [13] and Hashcat [4]. Hashcat is a CPU-based password recovery tool with another version named oclHashcat-plus which is GPU-based (or cudaHashcat-plus for NVidia graphical cards). Another version is Hashcat-lite for single hash attack. Due to the performance of GPU, we have chosen to test the deanonymization process with hashcat.

Hashcat supports 6 attacks mode, the software is multi-threads which allows the user to accelerate the hash speed considerably; the maximum speed for a powerful CPU is 87 M/s for 8 threads[6]. We are working with cudaHashcat-plus version, GPU-based, designed to work with NVIDIA graphic card. It supports most hashing algorithms as md5, SHA1, SHA512, MySQL. The software is optimized for dictionary attacks against multiple hashes. It has more options like supporting dictionary input from a pipe or taking multiple rule files. Different attacks exist with

[4]Organisational Unique Identifier
[5]The total numbre is increasing day per day, this value is for the 27th of july
[6]www.hashcat.net

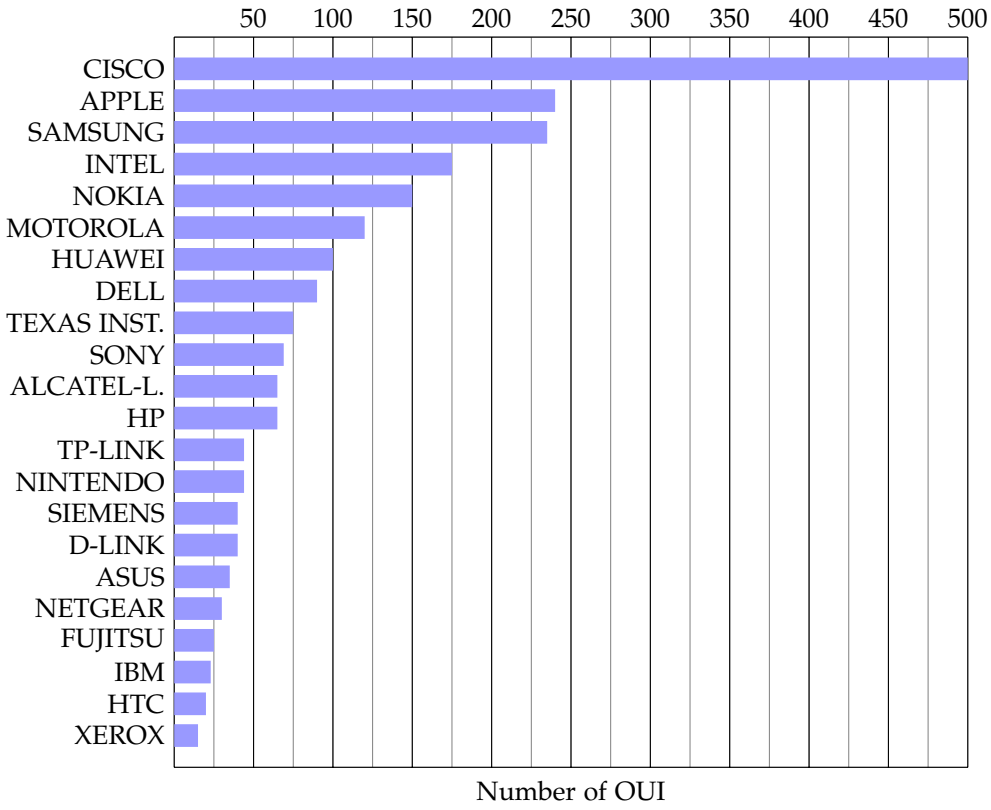

Figure 4: *Number of OUI per constructor*

this version, like the hybrid attack which is divided in 2 modes; one is a dictionary attack with suffix, the other with a prefix. The last software combined with cudaHashcat is maskprocessor. It is designed to generate dictionaries based on patterns supplied by the user. It can pipe output to cudaHashcat-plus for a brute-force attack.

V.2 Results

The test protocol is separated into 2 parts. First a test with CPU followed by a test with GPU. For the CPU-based test, the operating system is Ubuntu 12.10 (32 bits), the processor is an Intel(R) Core(TM)2 Duo CPU P9400 @ 2.40GHz,gcc version is 4.7.2. For the GPU-based test, the operating system is Ubuntu 12.04, the processor is an Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60GHz, the graphic card is a NVIDIA Corporation GF108 [Quadro 600].

The evolution of hardware performance let us think that bruteforcing all the possible addresses even if they are not all used should be feasible. The speed for bruteforce attack for the graphic card used in our experiment is 80 millions per second. The table 2 gives an overview of the time consumed to bruteforce all the MAC address range with different hardwares (taken from offical website).

With a CPU, the brute-force attack is possible but requires too much resources(296 days). But with the GPU, the amount of day decreases considerably to 41 for a single common graphic card. With the powerful GPU which reaches a cracking speed of 3,000 millions hashes tested per second,

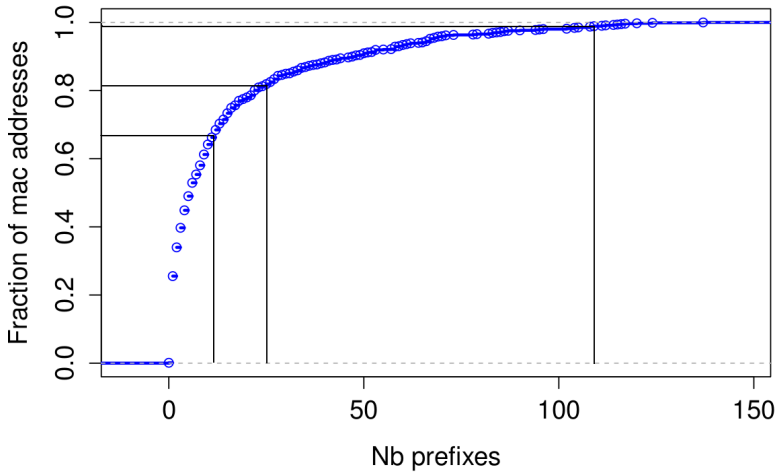

Figure 5: *Cumulative distribution of the prefixes in the considered MAC address dataset*

Hardware	Speed (M/s)	Time (s)	Time (h)	Time (day)
Integrated card (CPU)	11	25588634	7108	296
NVIDIA Quadro 600	80	3518437	977	41
NVidia gtx560Ti	433	650057	180	7.5
NVidia gtx570	629	447495	124	5
AMD hd7970	2136	131776	36	1.5
AMD hd6990	3081	91358	25	1

Table 2: *Cost of deanonymization for* 2^{48} *addresses (SHA1)*

the required time is only 25 hours. Moreover with parallelized GPU, the adversary is also able to decrease the time cost drastically.

Now, we take the benefit of the distribution found previously. Two experiments are made with CPU and GPU. The experiment is to understand the time spent to test 1000 digests with different amount of prefixes. Indeed, 10 prefixes should recover 60 % of the digests, 20 prefixes for 80% and 111 prefixes for 99%.

For the CPU, the optimized test is to save all the most prefixes used (found from our dataset) in a file as a dictionary (macPrefixList111), add a rule file (rulehexa6digits.rule) containing all the possible value for a prefix (0x000000 to 0xffffff -16 billions-) and run the test.The command is :

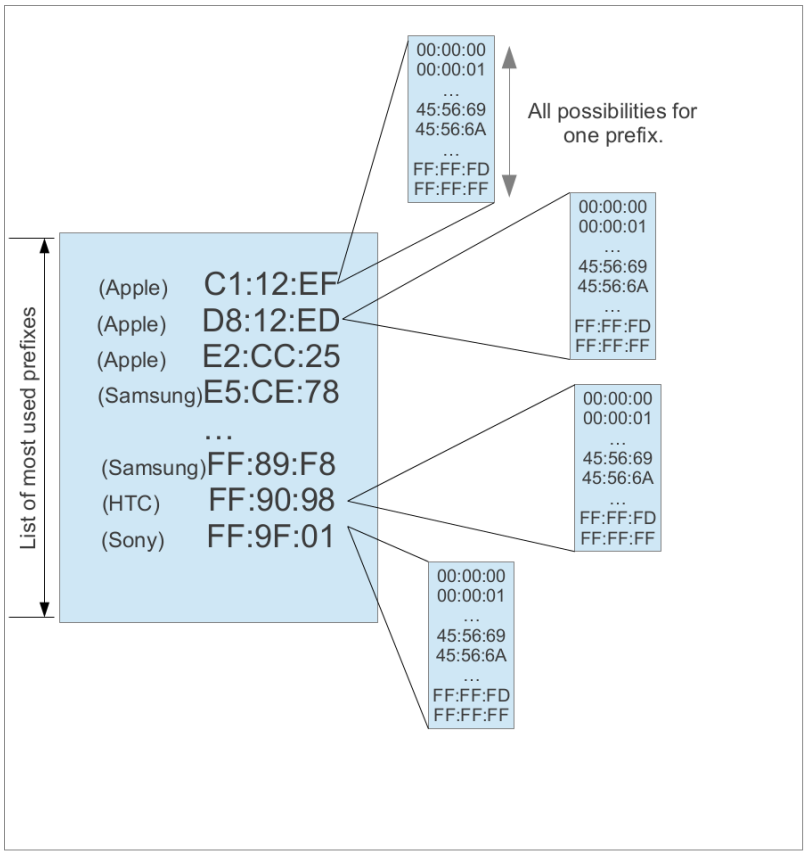

Figure 6: *Prefix-based attack. The left part corresponds to the most used prefixes and the right part corresponds to the suffix. Candidate MAC addresses are the concatenation of both.*

```
./hashcat-cli32.bin -a 0 -m 100 hashFileSHA1_1000 macPrefixList111
-r rulehexa6digits.rule -o found_hash
```

The figure 7 shows the time consumed by trying to deanonymize 1000 digests for three different amount of prefixes : 10, 22 and 111.

The figure shows that the applied algorithm is decisive, the impact on performance is considerable. For 10 prefixes, the time spent with SHA-512 algorithm is 50 times higher than with other algorithms. Now, hash tests will be calculated with sha1 algorithm, the figure 8 shows the time consuming for different amount of hashes. The possible number of MAC address correspond to $2^{24} \times Nb\_of\_Prefix$.

GPU tests are generally faster than CPU, however the user has to understand precisely how it works. Two parameters are interesting, the hash speed and the the GPU utilization. The same test elaborated for the CPU is made.

The command is :

```
./mp32.bin --custom-charset1 ?dabcdef ?1?1?1?1?1 | .././cudaHashcat-plus32.bin
-m 100 hashFileSHA1_1000  -r PRE_PREFIX22 -r theRule1digitHexa   -o found_hash
```

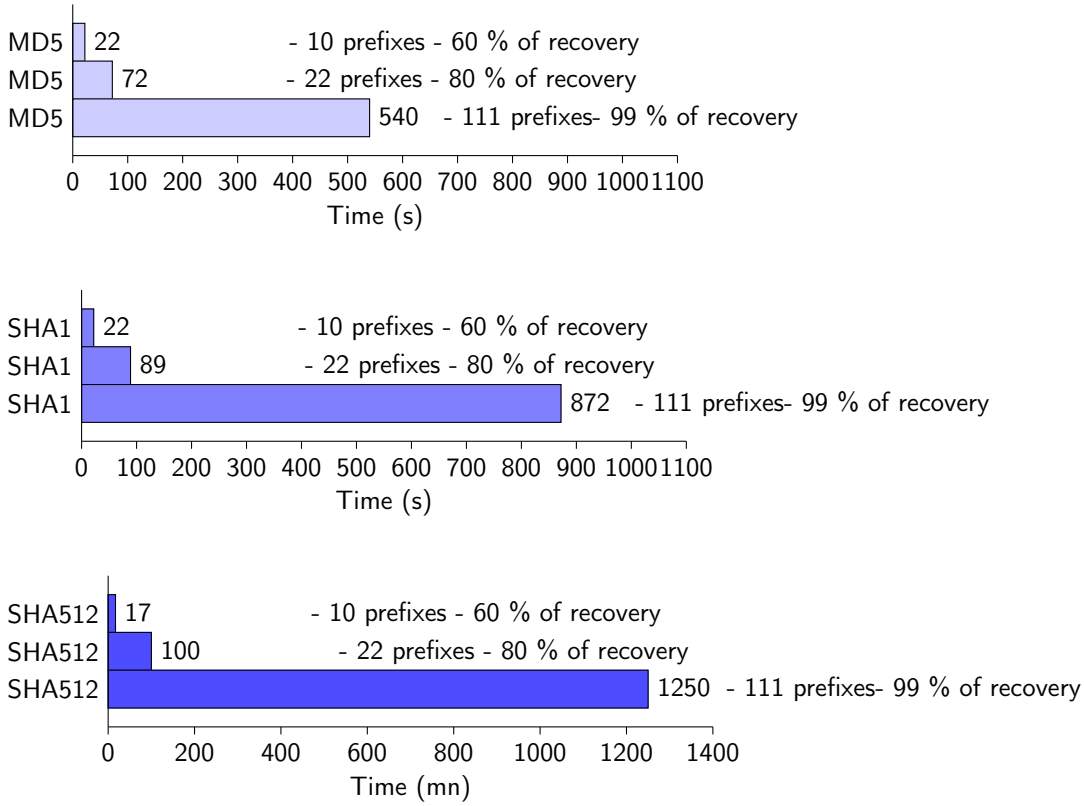

Figure 7: *Time spent for 1000 digests with 10, 22 and 111 prefixes*

The test consists in generating two rules, one for the prefix (PRE_PREFIX22) and one for the suffix (theRule1digitHexa). The prefix rule contains all the mostly used prefixes .

Then we are trying to brute-force the last 6 hexadecimal digits (containing all the possible MAC addresses for one prefix). The intuitive test is to create the suffix rule either by generating a wordlist with maskprocessor either by applying a mask.

However the GPU needs to have a massive amount of password candidates, that is why we have to optimize the GPU utilization. The mask would look like : ?*dabcdef*?1?1?1?1?1; *?d* means a numerical digit and *abcdef* means the extension to hexadecimal number. The optimization is to split the mask into two, three characters parts: ?1?1?1 and ?1?1?1. The left half of the mask will be given to maskprocessor to generate the candidate MAC addresses, while the right half will be used to generate the rules. This experiment is done with different sized left- and right-hand masks to achieve optimal GPU utilization.

The figure 9 shows the maximum cracking speed for 5 digits generated by maskprocessor and 1 for the GPU. This means that the trade off has been founded between the throughput of the CPU and the calculation power of the GPU. The performances of GPU are shown in figure 10.

Ten prefixes are tested with 1000 digests in 4s for SHA-1 algorithm. The time cost between SHA-512 and SHA-1 is only five times higher compared to the CPU performance. Retrieve more than 99 % of digests with 111 prefixes even with SHA-512 algorithm takes 4mn. This figure shows the potential of any adversary with a common hardware (a single GPU) to rapidly de-anonymize

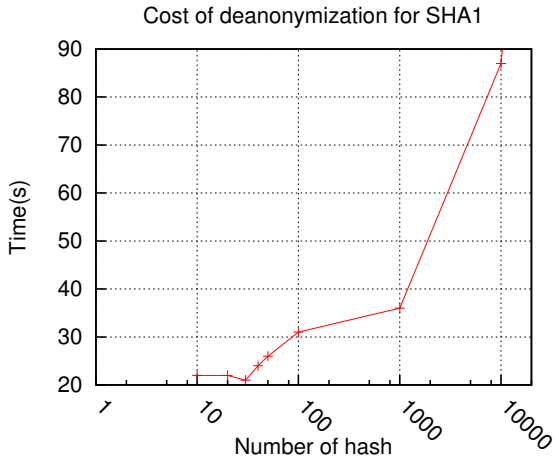

Figure 8: *Cost of deanonymization for SHA1*

a considerable amount of digests.

To conclude this part, hashing a MAC address is not a satisfactory solution. Several methods more or less fast allow an adversary to discover the hidden MAC addresses; the bruteforce attack with all range of 2^{48} addresses is possible with a efficient graphical card in only 25 hours. Moreover if the selection of prefixes is reliable, the work becomes very fast and last only a few minutes.

VI. Counter-measures

VI.1 Hashing multiple time

Because the application of a hash function on a data is note private, one can think to increase the data processing. Iterating n times the hash function (with n large) on the data can drastically increase computation time of the adversary. However, this has to be paid also while anonymizing the data. A possible solution to overcome this consists in using an anomyzation cache which stores a mac address and its corresponding hash value. Moreover, an adversary using special purpose hardware can get rid of this counter measure at the expense of hardware.

Cryptographic hash function are not the best tool to solve our problem. Key derivation functions and password storage are well-suited. Indeed; they are one-way as required and the implementation can not be speed up exploiting parallelism. Examples of such functions are bcrypt [21] and scrypt [20].

VI.2 HMAC : Keyed-hash message authentication code

The adversary is able to apply the same algorithm of encryption, one solution is to use a secret key unknown by the adversary. The principle of Message Authentication Code is to detect -when two parties communicate across an insecure channel- any attempt to modify the information or fake its origin. This was possible by adding to the message an authentication tag computed as a function of the data and the shared key [14]. Then the HMAC has been implemented with a cryptographic hash function which was used in a black box way so that it can be implemented with available code.

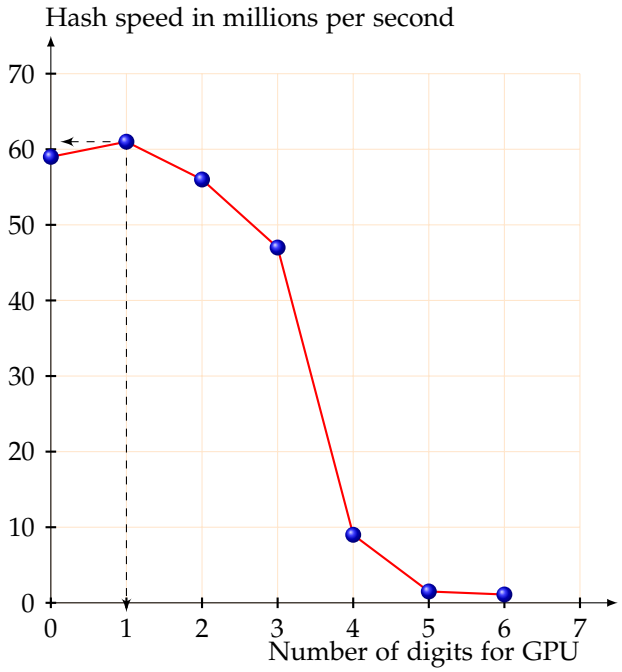

Figure 9: *The dependance of the workload of gpu on the hash speed*

A brief explanation, let H be the hash function, it takes any length of input and output l-bit (l = 128 for MD5 and l = 160 for SHA-1). Let *some_text* be the data and K the Message Authentication secret key shared by the two parties. Then we define two fixed and different 64 bytes strings ipad and opad. The function HMAC has two arguments K and *some_text* and produces $HMAC_K(some\_text) = H(K \oplus opad, H(K \oplus ipad, some\_text))$

The advantage of this technique is that the overall computational cost is close to the cost of a single hash. In addition, the code of H is not modified, that means that if the the hash function is compromised, the evolution to a secure other hash function is possible easily. However the problem about the key storage is not solved. Because the principle of physical tracking system is based on the recognition of a discovered device in the past, the key needs to remain the same for a monitoring period. It also could not be stored next to the digests, in a case of leak, the recovering process would be done easier. One idea is to store the key in a different place, it could be directly in nodes memory. Before sending the recoded MAC address, the secret key could be concatenated to the MAC address and so the message will be composed of the key + MAC. This solution is also limited, indeed if an adversary compromised only one node all the system will be compromised. One can think other solutions about the secret-key storing place. To conclude, the HMAC solution is adapted to publish the database because the key is forgotten but for a permanent working the key storage issue has to be taken in consideration.

VI.3 Setting a Wi-Fi based tracking system with Raspberry Pi

The last step of this research is to set a real tracking system with Raspberry Pi. The experiment consisted of placing the sensors in different places to cover a precise area, generally crowded place. The hardware used is a mesh of Raspberry Pi. This device is a cheap (35 €) and small computer.

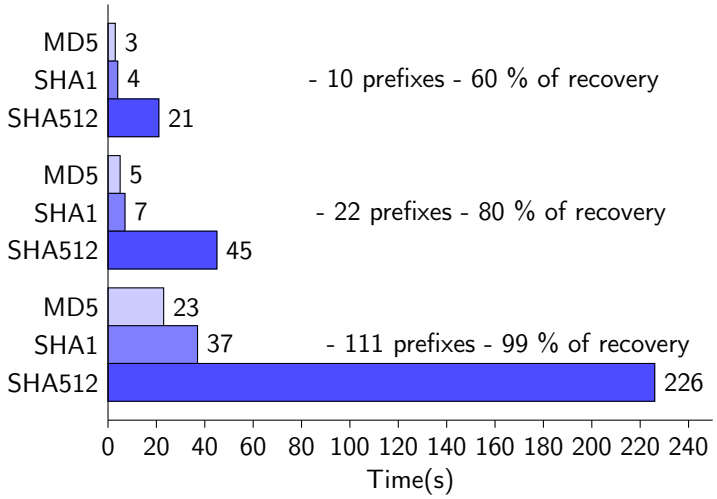

Figure 10: *Time for 1000 digests with 10, 22 and 111 prefixes for GPU*

The low power consumption and small dimension make it very useful for several applications. The Figure 11 shows an equipped machine for the experiment. All the devices are provided with a wired network connection, a Wi-Fi dongle supporting monitoring mode to capture packets. The OS installed in the SD card is Raspbian, a specific Debian-based GNU/Linux distribution.

Figure 11: *An equipped Raspberry Pi*

The figure 12 shows the experiment. The client-server communication is based on a TCP/IP socket, the MySQL database is storing all the received packets. A special software filter has been developed to aggregate and count the sames frames sent in a period of 2 or 3 seconds. For example a probe request is sometimes sent from one STA 10 times in a few ms. On client, the packets are monitored and aggregated and then sent to the server. On the server, data is first protected and stored in the database. Next, statistics are computed.

The data could be used to monitor a precise area, counting the number of individuals that are visiting or crossing this area. The figure shows 13 a example of use for a day-long monitoring. We notice the arrival and departure times. The capture coverage is including the cafeteria which is opened from 9:15 to 11:30 and from 12:15 to 14:00. The graph variation is confirming those opening hours. The next step is to distinguish if the visitor is just crossing the area or is staying

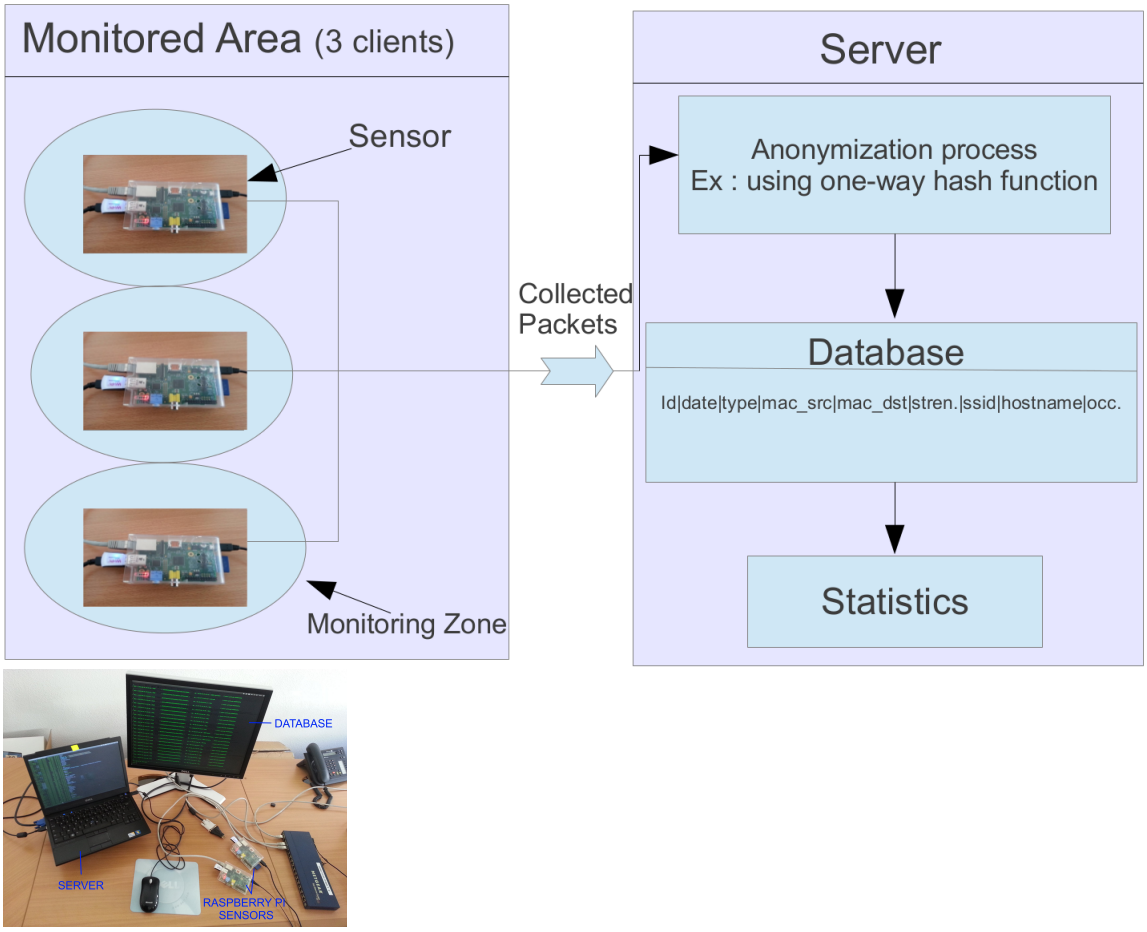

Figure 12: *Experiment*

more (in this case how many times). The test platform is new that is why we do not have more results. But the future results are promising.

VII. Conclusion and future work

We have shown that the de-anonymization of a protected dataset is possible thanks to two attacks. One is the bruteforce attack which can take a long time, the other is an attack based on the MAC addresses distribution where de-anonymizing a set of thousands protected MAC addresses takes less than one minute. We notice indeed that the most used devices are gathered in a few constructor. Different solutions have been presented as the multiple hashing or the HMAC, they are limited by the resources cost or the architecture of the system. The results lead us to understand that there is a tradeoff between the protection of privacy and feasability of a long-period tracking system.

It would be interesting to apply the proposed methods to other similar situations because the anonymity of unique identifier is not restricted to MAC addresses. One similar case is the IMEI number which identifies a mobile phone. Some applications of smartphones are using

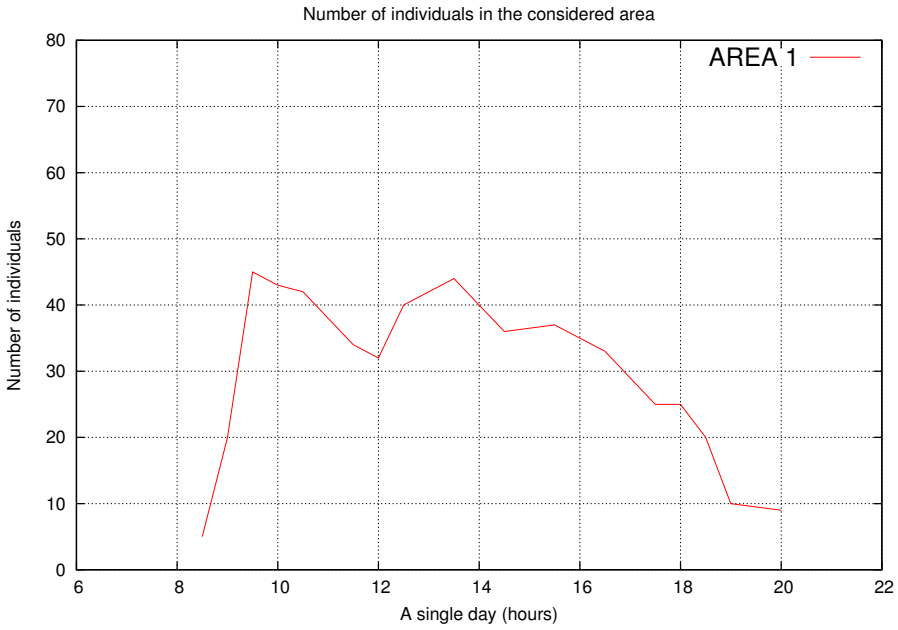

Figure 13: *Number of detected devices*

IMEI[7] as unique identifier, the security is a digest with simple MD5 hash function, this means easy-to-compute and fastly de-anonymized.

Finally, the data acquired in the experiment are limited to a few days. The next step is to have more data to simulate a real-case of tracking and analyze how the protection solutions will work.

[7]International Mobile Equipment Identity

Glossary

- **AP : Access Point**, is a device, such as a wireless router, that allows wireless devices to connect to a network.
- **IMEI : International Mobile Station Equipment Identity**, is a number, usually unique to identify GSM.
- **MAC : Media Access Control**, is a unique identifier assigned to network interfaces for communications on the physical network segment.
- **SSID : service set identifier**, is a unique ID that consists of 32 characters and is used for naming wireless networks. When multiple wireless networks overlap in a certain location, SSIDs make sure that data gets sent to the correct destination.
- **MD5 :**, The MD5 message-digest algorithm is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value.
- **SHA1 : Secure Hash Algorithm**, SHA-1 is one of several cryptographic hash functions, most often used to verify that a file has been unaltered.
- **CPU : Central Processing Unit**,is the brains of the computer where most calculations take place. In terms of computing power, the CPU is the most important element of a computer system.
- **GPU : Graphics Processing Unit**, is a single-chip processor. However, it is used primarily for computing 3D functions. This includes things such as lighting effects, object transformations, and 3D motion.

References

[1] http://standards.ieee.org/develop/regauth/oui/public.html.

[2] http://www.euclidanalytics.com.

[3] http://www.libelium.com/.

[4] http://www.linuxjournal.com/content/hack-and-password-cracking-gpus-part-ii-get-cracking.

[5] http://www.navizon.com/.

[6] Alessandro Acquisti. Privacy and security of personal information. In L.Jean Camp and Stephen Lewis, editors, *Economics of Information Security*, volume 12 of *Advances in Information Security*, pages 179–186. Springer US, 2004.

[7] Paul C. van Oorschot Alfred J. Menezes and Scott A. Vanstone. In *Handbook of applied cryptography*, page 64, 2001.

[8] Ashar Aziz and Whitfield Diffie. Privacy and authentication for wireless local area networks. *Personal Communications, IEEE*, 1(1):25–31, 1994.

[9] John Black, Martin Cochran, and Trevor Highland. A study of the md5 attacks: insights and improvements. In *Proceedings of the 13th international conference on Fast Software Encryption*, FSE'06, pages 262–277, Berlin, Heidelberg, 2006. Springer-Verlag.

[10] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in Wi-Fi probe requests. *Pervasive and Mobile Computing*, (0):–, 2013.

[11] Philip Hawkes, Michael Paddon, and Gregory G. Rose. Musings on the wang et al. md5 collision. Cryptology ePrint Archive, Report 2004/264, 2004. `http://eprint.iacr.org/`.

[12] Vlastimil Klima. Tunnels in hash functions: Md5 collisions within a minute. Cryptology ePrint Archive, Report 2006/105, 2006. `http://eprint.iacr.org/`.

[13] Ryan Lim. Parallelization of john the ripper (jtr) using mpi. *Nebraska: University of Nebraska*, 2004.

[14] R. Canetti M. Bellare and H. Krawczyk. Message authentication using hash functions: The HMAC construction. In *RSA Laboratories' CryptoBytes*, pages Vol. 2, No. 1, 1996.

[15] Mohamed Ali Kaafar Mathieu Cunche and Roksana Boreli. I know who you will meet this evening. *IEEE WoWMoM 2012*, June 2012.

[16] A. B. M. Musa and Jakob Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 281–294, New York, NY, USA, 2012. ACM.

[17] Arvind Narayanan. Fast dictionary attacks on passwords using time-space tradeoff. In *In ACM Conference on Computer and Communications Security*, pages 364–372. ACM Press, 2005.

[18] Navizon. Measuring crowds or people counting: What is the difference?, June 2012.

[19] Yuval Ofir Omri Ildis and Ruby Feinstein. Wardriving from your pocket. 2013.

[20] Colin Percival. Stronger Key Derivation via Sequential Memory-Hard Functions. In *USENIX Annual Technical Conference, FREENIX Track*, Ottawa, Canada, 2009.

[21] Niels Provos and David Mazières. A Future-Adaptable Password Scheme. In *USENIX Annual Technical Conference, FREENIX Track*, pages 81–91, Monterey, California, USA, 1999. USENIX.

[22] Ronald Rivest. The md5 message-digest algorithm. 1992.

[23] Yu Sasaki and Kazumaro Aoki. Finding preimages in full md5 faster than exhaustive search. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 134–152. Springer Berlin Heidelberg, 2009.

[24] Bill Schilit, Jason Hong, and Marco Gruteser. Wireless location privacy protection. *Computer*, 36(12):135–137, 2003.

[25] Xiaoyun Wang, YiqunLisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer Berlin Heidelberg, 2005.