Analysing the privacy policies of Wi-Fi trackers

Levent Demir, Mathieu Cunche, Cédric Lauradoux

Analysing the privacy policies of Wi-Fi trackers

Levent Demir, Mathieu Cunche, Cédric Lauradoux

RESEARCH REPORT

N° 8506

March 2014

Project-Team Privatics

ISSN 0249-6399   ISRN INRIA/RR--8506--FR+ENG

Analysing the privacy policies of Wi-Fi trackers

Levent Demir, Mathieu Cunche, Cédric Lauradoux

Project-Team Privatics

Research Report n° 8506 — March 2014 — 14 pages

Abstract: Wi-Fi-based tracking systems have recently appeared. By collecting radio signals emitted by Wi-Fi enabled devices, those systems are able to track individuals. They basically rely on the MAC address to uniquely identify each individual. If retailers and business have high expectations for physical tracking, it is also a threat for citizens privacy. We analyse the privacy policies used by the current tracking companies then we show the pitfalls of hash-based anonymization. More particularly we demonstrate that the hash-based anonymization of MAC address used in many Wi-Fi tracking systems can be easily defeated using of-the-shelf software and hardware. Finally we discuss possible solutions for MAC address anonymization in Wi-Fi tracking systems.

Key-words: Wi-Fi tracking, MAC address, 802.11, anonymization, privacy, hash function

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Analyse des politiques de protections de la vie privée des systèmes de traçage Wi-Fi

Résumé : Les systèmes de traçages basés sur le Wi-Fi ont récemment fait leur apparition. En collectant les signaux radio émis par les terminaux équipés du Wi-Fi, ces systèmes sont capables de tracer les individus. Ils utilisent l'adresse MAC des terminaux pour identifier de manière unique les personnes. Si les acteurs du commerce physique ont de grandes attentes de ces technologies de traçage physique, elles représentent également une menace pour la vie privée. Nous analysons les politiques de vie privée des principaux acteurs du traçage Wi-Fi et nous montrons l'inefficacité des techniques d'anonymisation par fonction de hachage. Plus particulièrement, nous montrons que les techniques d'anonymisations basées sur les fonctions de hachage, communément utilisés dans les systèmes de traçage Wi-Fi, peuvent être facilement cassées en utilisant des logiciels et du matériel standard. Finalement, nous discutons des solutions alternatives pour l'anonymisation des adresses MAC dans les systèmes de traçage Wi-Fi.

Mots-clés : Traçage Wi-Fi, adresse MAC, 802.11, anonymisation, vie privée, fonction de hachage

1 Introduction

Knowing human dynamics such as the people path, the crowd size or the visit duration and frequency are extremely valuable information for many applications. It offers great prospects to retailers or for urban planning. Gathering location analytics also known as tracking was done using visual census, mechanical/optical systems or processing of CCTV streams [23]. Wi-Fi-enabled portable devices changed everything. They broadcast periodically a unique identifier in the clear. By collecting this identifier it is possible to detect individual, triangulate their position and track their movements. Several Wi-Fi tracking systems are already deployed in retail places where they provide information on customers or on road where they provide insight on traffic [5, 8, 6].

If Wi-Fi tracking systems provide invaluable information for retailers, they are a clear threat to individuals' privacy. By recording the whereabouts of any individual that happen to carry a device with Wi-Fi turned on, they can monitor the activities of a large fraction of the population. They do not need any consent of the the user and are totally passive. Therefore, it is impossible for the user to know if whether or not tracking is performed.

Wi-Fi trackers in response to citizens concerns have adopted privacy policies to reduce the privacy risks. Those privacy policies describe method employed by Wi-Fi trackers to securely manage the private information within Wi-Fi tracking systems. In this paper, we review the privacy policies of 15 major Wi-Fi tracking companies. A key feature of those privacy policies is to anonymize the MAC address using a hash-function.

We demonstrate that hash-based anonymization is weak and that MAC addresses can be recovered using an appropriate guesswork. More particularly, using a real world dataset of MAC addresses, we show that hashing can be inverted in a matter of minutes.

Finally we present the possible countermeasures and discuss their constraints related to their integration in Wi-Fi tracking systems. We discuss the limitation of naive approaches such as the addition of a random value in the anonymization process. Then, we propose an practical anonymization solution based on encryption an hash-chain.

This document is organized as follows. Section 2 describes Wi-Fi tracking systems. Thei trackers privacy policies are reviewed in Section 3. Section 4 presents how hash based anonymization failed to protect MAC addresses. Section 5 discuss solutions for private storage of MAC addresses and Section 6 concludes the paper.

2 Wi-Fi tracking systems

Wi-Fi tracking systems are keeping track of people whereabouts using the messages broadcasted by their Wi-Fi enabled device [28]. Indeed Wi-Fi devices and especially portable ones use an active service discovery mechanism to search surrounding access points [20]. In active service discovery mode, a device periodically broadcasts probe request frames. Upon reception of a probe request, an access point (AP) replies with a probe response, thus declaring its presence. For energy-saving reasons, the active service discovery method is preferred to the passive one, in which a device passively listen to beacon frames emitted by the APs.

Probe requests are broadcasted over Wi-Fi channels without any encryption, leaving their content available. Amongst other information [25], those frames contain the MAC address of the emitting device. The MAC address is a 48-bit identifier uniquely allocated to a device. Therefore, when using the active service discovery mode, a Wi-Fi enabled device is periodically broadcasting a unique identifier that can be used to track the owner of the device.

Wi-Fi tracking systems [28] are composed of sensors deployed over an area of interest and a server in charge of centralizing and storing the information collected by the sensors (see Figure 1).

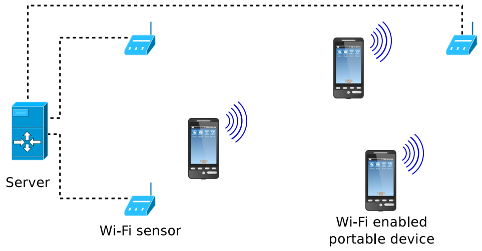

Figure 1: Architecture of a Wi-Fi tracking system.

When a device comes in range of a sensor, it will be detected thanks to the probe requests it emits. Sensors are collecting the information contained in probe requests. For each received probe request, the system records the source MAC address, the time-stamp, and the identifier of the sensor that have recorded the probe request. From this information, the system can deduce the presence and the path of a Wi-Fi device across the area covered by the sensors. The format of a typical entry in the database is the following:

`<time> , <MAC_address> , <location>`.

When the density of sensors is high, probe requests can be overheard by several sensors. In this case, an accurate location of the device can be computed based on the RSSI (Received Signal Strength Indicator) received by the corresponding sensors. This is done by triangulating the position of the source from the signal strength and the location of the sensors. A device triangulates its geolocation from surrounding cell towers or Wi-Fi APs in the same way.

The main purpose of Wi-Fi tracking systems is to monitor the human activity in physical spaces. Wi-Fi tracking systems are currently used to monitor the road for urban planning. By capturing Wi-Fi signal of devices aboard cars, Wi-Fi tracking systems can efficiently detect traffic congestion and compute point-to-point travel time [8]. Another popular application is physical analytics in retail places [5, 6]. Wi-Fi tracking systems collect information on the customers path within a retail places and are able to extract information about visitors habits. This involve statistics such as the number of visitors, the length and frequency of their visit, or their dwell time.

In order to enable the computation of statistics required by physical analytics application, the system must be able to uniquely identify each device. This is done thanks to the MAC address of each device that is by definition a unique identifier. Therefore, any other identifier could be used in place of the MAC address as long as it is also unique.

3 Privacy policies

During the last few years, a number of companies providing Wi-Fi tracking systems and services have appeared. Aiming at reducing the impact on the privacy of the individuals tracked by those systems, each company has adopted its own privacy policy. These privacy policy are described on the website of those actors under the form of a privacy statement or as part of the FAQ (Frequently Asked Questions).

More recently 10 major Wi-Fi tracking actors[1] along with the Future of Privacy Forum (FPF) have formed a working group on the privacy aspects of the technology. The first outcome of this

[1] The company *PathIntelligence* is not included in our study because it used GSM signals instead of Wi-Fi

project is the creation of a document entitled "Mobile Location Analytics Code of Conduct" [29] describing guideline for privacy protection.

To better understand the privacy policies adopted by Wi-Fi trackers, we have selected 15 Wi-Fi tracking companies, including the 10 companies involved in the FPF initiative.

Table 1: Description of the privacy policies for major Wi-Fi tracker († indicates members of the FPF initiative).

Company	Data collection	Data transfer	Data anonymization	Data storage	Retention	Opt-out
Aislelabs [1] †	MAC, RSSI	SSL	*Randomization and one-way hashing*	Third party providers	24 months	Yes
Brickstream [2] †	MAC, RSSI	-	*"hash" or scramble*	-	-	Yes
Euclid [3] †	MAC, RSSI, manufacturer	SSL	*Hashing*	Amazon Web Services	24 months	Yes
eyeQ [4] †	-	-	-	-	-	Yes
iInside [7] †	MAC	*Secure connection*	*Assigning the signal ID a random code*	-	-	Yes
Measurance [10] †	-	-	-	-	-	No
Mexia [11] †	MAC, RSSI, manufacturer	*256-bit (SHA-2) encrypted connection*	*Encryption with the highest standards in the industry*	Rackspace	-	Yes
Radius Networks [14] †	MAC	-	*One-way hashing*	-	30 days	Yes
Solomo [17] †	-	-	-	-	-	No
Turnstyle [18] †	MAC	-	SSL	*secure server*	-	No
ReadMe Systems [15]	MAC, RSSI	SSL	*Randomization and one-way hashing*	*enterprise-class physical and network security*	-	Yes
RetailNext [16]	MAC, RSSI	-	-	*Third party service providers*	30 months	Yes
Nomi [13]	MAC, RSSI, manufacturer	-	*Hash function*	*Third party service providers*	18 months	Yes
Walkbase [19]	MAC	-	*Hash function*	-	-	Yes
Navizon [12]	MAC, RSSI	-	-	-	-	No

Based on the MLA (Mobile Location Analytics) code of conduct [29] and the privacy policies of the considered Wi-Fi tracking companies, we have identified five critical steps in the data management process of Wi-Fi tracking systems: data collection, transfer, anonymization, storage and opt-out mechanism. For each step, Wi-Fi tracking companies have have adopted various measures to reduce the privacy risk. The details of those measures are presented in Table 1. They can be summarized as follows:

Data collection: most privacy policies aim at enforcing data minimization : the data collected are kept as minimal as possible : the MAC address is often collected and in some cases the signal strength and the manufacturer. Some companies emphasized that no other information such as name, browsing history or e-mail address, are collected.

Data transfer this is the second phase of the data process during which the information collected by sensors is sent over to a central server. The confidentiality of the transferred information must be guaranteed. Some privacy policies specify that the data is securely transmitted, presumably over a secure channel, while other go in more details and mention SSL as the technology employed to implement this secure channel. We note that a majority of the privacy policies does not mention the security of this phase.

Data anonymization most privacy policies acknowledge that the MAC address, is a sensitive piece of data that must be transformed in such a way that it is not possible to recover the original value. To perform this task, a number of privacy policies mention the use of a *hash function* or a *cryptographic hash function*. In addition to those references, the usage of hash function is also mentioned by the FPF code of conduct [29] as a mean to 'De-Personalize' data. Other methods, described as *encryption* and *randomization* are mentioned. We note that in many cases the privacy policies lack details about characteristic of the hash or encryption function used as well as the details of the anonymization process.

Data storage the data stored by mobile tracking systems can be seen as individual mobility traces and its confidentiality must therefore be preserved. For many companies, data are stored on Cloud computing platforms with some of them relying on the security guarantees of the cloud provider. In other cases, encryption of the data is also mentioned, but there is a lack of details on where is the key material stored and how it is managed. In addition, an other interesting element concerning storage is the data retaining time. It is only mentioned by 5 out of 15 companies and range from 30 days for Radius Networks up to 30 months for RetailNext. Even if data retention statement is part of the FPF code of conduct [29], most of the subscribers to this code are not mentioning it in their privacy policy.

Opt-Out finally, some Wi-Fi tracking companies have an Opt-Out mechanism in which individuals not willing to be tracked can enter their Wi-Fi and Bluetooth MAC addresses. By doing so data concerning those MAC addresses will be removed and no other data will be collected on them. To ease the Opt-Out task, the FPF group has create centralized Opt-Out mechanism that comes with indication on how to find the MAC address of a device: `https://optout.smartstoreprivacy.org/`. The effective date after which the Opt-Out request is enforced is not always specified, but Aislelabs is ensuring the disassociation of the previous collected data from the MAC within 30 days and Euclid within 7 days. A total of 4 companies do not mention the Opt-Out mechanism in their privacy policies, even if for 3 of them, the centralized Opt-Out [29] website cite them as involved.

Overall, we found only 2 out of 15 companies for which the privacy policies are covering all the previously mentionned steps. This does not mean that those steps are not included, but this give an idea of the importance of this issue from the company point of view. Furthermore, when covered, the technical details used to enforce a particular step are rarely provided or are vague. Finally, as we will see in the following sections, the hash-based anonymization, used by a number of companies to anonymize data, is not sufficient to guarantee the anonymity of the MAC address.

4 Hashing MAC addresses

In this section, we first evaluate hash-based anonymisation, the method used by several trackers to anonymize MAC addresses in their databases. Based on a real-world dataset, we demonstrate that the MAC addresse transformed with this method can be easily re-identified.

Let us assume that the database has been leaked. The critical question is the difficulty to recover a MAC address from its digest. The case we just describe is very similar to an `/etc/shadow` file being leaked to a hacker. Recovering passwords or MAC addresses are indeed the same problem. Password cracker such as John the Ripper [24] or hashcat [32] can be used. In our work, we have used hashcat.

To assess the security of a digest, we need to know the cost of testing all the possible 2^{48} MAC addresses by brute force search. Our tests were performed on a computer running Windows 8.1, equipped with an Intel core i5-2500k processor and an ATI R9 280X graphic card. In order to exploit the computational power of the graphic card, we used the *oclHashcat* flavor of *hashcat*, which support ATI GPUs acceleration. We have also used the benchmarks provided by hashcat [32] to give some extrapolations. The results are given in Table 2 for the hash function SHA-1. On our laptop it takes, 296 days while a computer equipped with an AMD Radeon HD6990 can run the whole search in only a day.

Table 2: Computation time for 2^{48} SHA-1 digests. (* means our experiments)

Hardware	Speed (M/s)	Time (days)
Integrated card (CPU)*	11	296
NVIDIA Quadro 600	80	41
NVIDIA GTX 560 Ti	433	7.5
NVIDIA GTX 570	629	5
ATI R9 280X*	1228	2.6
AMD HD 7970	2136	1.5
AMD HD 6990	3081	1

This demonstrates that the hash-based anonymization used in Wi-Fi tracking systems can be defeated using a free on-the-self software with a high-end GPU and with an overall cost lower than 2000 dollars. Such a result is enough to conclude that this anonymization method is not sufficient to answer MAC address privacy. We need to push this attack to its limit in order to find the appropriate counter measure.

MAC addresses of Wi-Fi network interfaces are not evenly distributed amongst the set of possible values. The Figure 2 describes the structure of a MAC address. MAC addresses are allocated to vendors by range of 2^{24}. Each range is identified by a OUI prefix that corresponds to the left part of the MAC address. The remaining 24-bit of the MAC are the network interface controller (NIC) which identifies a interface within a given OUI range. So far only 0.1% of the OUI prefixes have been allocated [9], meaning that in the wild their are $\approx 2^{38}$ different MAC addresses. This first observation already gives us ×1024 speedup. It can be improved further.

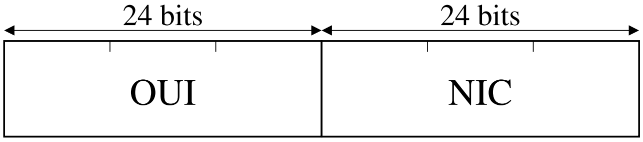

Figure 2: Structure of a MAC address.

The Figure 3 presents the number of OUI prefixes allocated for the top vendors. In the space of allocated MAC addresses, only a fraction corresponds to Wi-Fi interfaces. Indeed, MAC addresses are given to many type of interfaces (Ethernet, ATM, Bluetooth, etc.) and Wi-Fi interfaces represent a subset of all the existing network interfaces. The oldest OUI prefixes allocated are unlikely to match Wi-Fi devices. Therefore, the set of possible values for a MAC address corresponding to a Wi-Fi interface can be narrowed down.

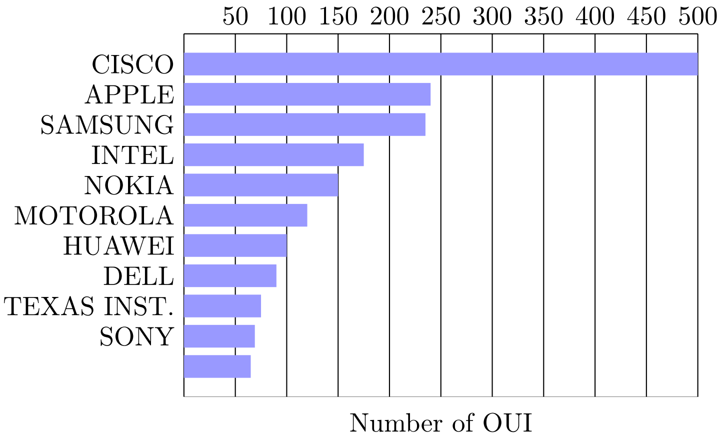

Figure 3: Top 10 constructors in terms of OUI.

We have studied the distribution of Wi-Fi interfaces' MAC addresses in a real world dataset containing more than 15.000 MAC addresses. This dataset has been obtained following the same protocol used by Wi-Fi tracking systems, i.e. by monitoring wireless channels to collect MAC addresses of Wi-Fi enabled devices. More specifically, a laptop equipped with a Wi-Fi interface placed in monitoring mode has been carried around in public places as in [22]. The captured traffic was strictly limited to probe request frames, the same frames that are collected by commercial Wi-Fi tracking systems [5]. From those captured frames, we only kept the source address field that contains the MAC address of the emitting device, no additional information such as timestamps or location was recorded. Once collected the data has been aggregated per OUI and the original MAC addresses have been erased. This dataset can be considered as a representative sample of MAC address that can be found in Wi-Fi tracking systems, since we used the same method to collect the data.

Amongst the 15.000 addresses, we find only 859 different OUI prefixes. In practice, it implies that an exhaustive search costs only $\approx 2^{34}$ ($\times$16384 speedup over the naive search).

The Figure 4 shows the cumulative distribution of the MAC address prefixes. It shows that a majority of the MAC addresses found in our dataset are concentrated under a small number of prefixes. More particularly, 87 prefixes covers 50 % of the dataset while 361 prefixes covers 90 % and 709 covers 99%. Instead of assuming that all the OUI prefixes have the same probability of occurrence, we can exploit their distribution to speedup the search. We enumerate the OUI prefixes in order of decreasing probability. We implement in this way a guesswork [21, 27]. In average, this guesswork costs $\approx 2^{30}$ ($\times$262144 speedup) and is dependent of the distribution of the MAC.

The Figure 5 shows the time consumed to make an α-guesswork [21], *i.e.* recovering α% of the MAC addresses from their digests. We consider a set of 1000 digests and the values 50%, 90% and 99% for α. The re-identification of the MAC addresses is done most of the time in a matter of minutes even for the costly SHA-512 function.

Finally, we note that some Wi-Fi tracking systems store the name of manufacturer along with the hashed MAC address (see Table 1). This information could be exploited by an attacker to

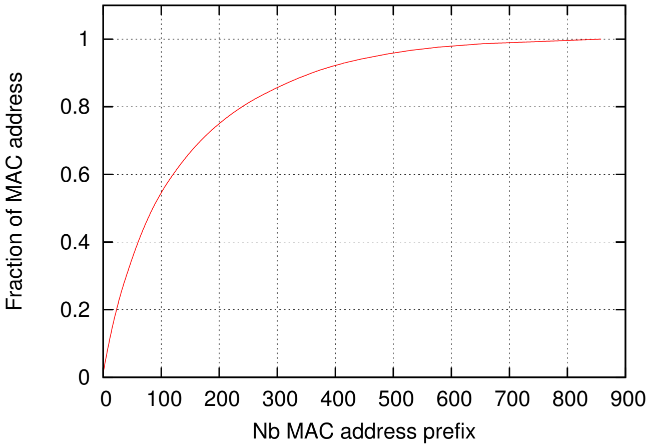

Figure 4: Cumulative distribution of OUI prefixes in the sample dataset.

further improve the speed of the attack, by reducing the search to the OUI ranges corresponding to the manufacturer.

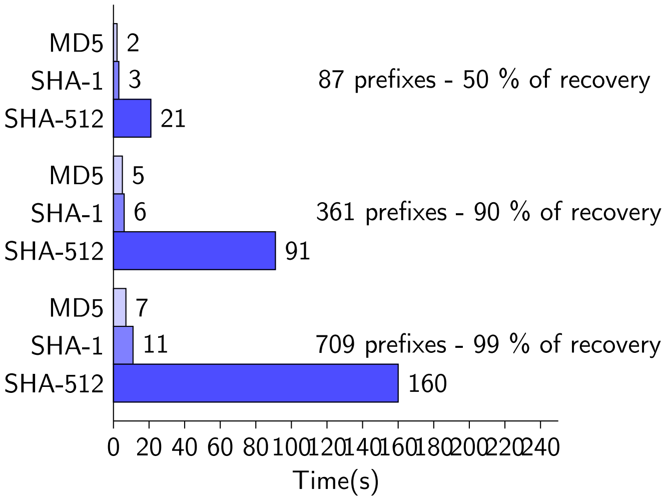

Figure 5: Re-identification of 1000 hashed MAC addresses using 87, 361 and 709 OUI prefixes (using *oclHashcat-plus* on a ATI R9 280X GPU).

5 Storing MAC privately

The problem we face is very similar to passwords storage. A first solution could be to increase the guesswork cost by using `bcrypt` [31] or `scrypt` [30]. Compared to `SHA-512`, we observe a $\times 21.10^3$ slowdown using `bcrypt` on hashcat. This solution would be only temporary due to the

Moore's Law. In the long term, an adversary will have enough computational power to recover the MAC address from a digest. In addition, this approach will induce an overhead for the Wi-Fi tracking system, since the cost of anonymization process will also be increased.

Randomness The entropy of MAC addresses is not high enough to prevent a guesswork after hashing. The obvious solution consists in increasing the input entropy by introducing random values. There are two strategies to do so. The first strategy consists to append a random value r to the MAC address x before hashing: $H(x||r)$. In the most extreme case, a MAC address is directly replaced by a random value which would correspond to UUID version 4 [26]. After doing so, the association between the random value and the MAC address is forgotten: it means that if the address reappeared later it will associated another random value. Let us assume that the random value is ℓ-bit long. The guesswork of the adversary is at least $O(2^\ell)$. The drawback of this solution is that the companies can not link the different entries of the database to a given address any more. They only view connection events. However, this is the best solution for users privacy.

The alternative consists to introduce a ℓ-bit secret key. From now, we use a secure block cipher with encryption E_K and decryption D_K using key K. It is also possible to use a keyed hash function. The AES is the obvious choice but the recent lightweight block ciphers are also of interest. A MAC address is now replaced in the database by $E_K(\text{MAC})$. Without the knowledge of the key, the adversary needs to guess the key and the MAC address. He needs to do $O(2^\ell)$ operations at least. In our model, the adversary can compromise the storage server. If this entity performs the anonymization, the adversary knows the key and can re-identify the MAC addresses. If the key is changed for each entry of the database, the adversary can only learn re-identify the last entry. However, the company can not link the entries associated to a given address.

Fortunately, there is a trade-off between these two previous extremes. The key used to encrypt the MAC addresses is obtained from a chain of hash as shown in Figure 6. The the chain is the master key K. This key is used once by the storage server and it is known by the entity computing the statistics. For each entry, a new key is generated using a cryptographic one-way function and the previous key is erased. This solution has three advantages. First, an adversary cannot link the entries of the database without the knowledge of the root of the whole chain: the same address is encrypted several times but under different keys. Second, an adversary getting the control of the storage server would only recover the last entry. The previous keys have been erased and it is difficult from the last key to recover the previous one. Finally, recovering each entry sequentially requires only the computation of a cryptographic hash function and a block-cipher. Getting a random entry i, requires i computation of the hash function.

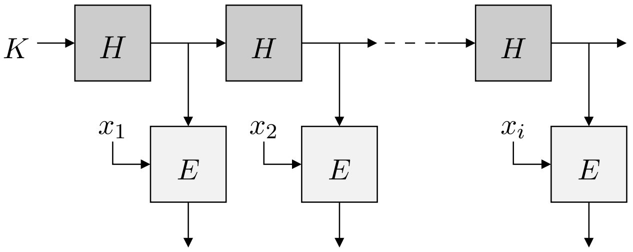

Figure 6: Anonymization using a hash chain and a block-cipher for a sequence of addresses $(x_1, x_2 \cdots x_i)$.

6 Conclusion

In this paper we discuss the privacy policies of Wi-Fi trackers. The existing description of these policies are far too evasive to inspire trust of the citizens. Moreover, hash-based anonymization used by some companies is clearly too weak: it can be broken in a matter of minutes. We then present potential countermeasures to this attacks and discuss the constraints associated to their implementation in Wi-Fi tracking systems. Privacy preserving opt-out have not been treated in our paper. It is a critical topics left for future work.

References

[1] Aislelabs - Privacy. http://www.aislelabs.com/privacy/.

[2] Brickstream - Privacy Matters. http://brickstream.com/live/privacy-matters/.

[3] Euclid Analytics - Privacy Statement. http://euclidanalytics.com/privacy/statement/.

[4] eyeQ - Privacy Policy. http://www.eyeqinsights.com/?page_id=183.

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

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

[7] iInside - Privacy Policy. http://iinside.com/privacy-policy/.

[8] Libelium - internet of things platform provider. http://www.libelium.com/.

[9] Ma-l public listing. http://standards.ieee.org/develop/regauth/oui/public.html.

[10] Measurence. http://www.measurence.com.

[11] Mexia - Privacy Certified. http://mexia.ca/privacy-certified/.

[12] Navizon - Privacy. http://support.navizon.com/are-there-any-security-and-data-privacy-concerns-when-using-navizon-products/.

[13] Nomi - Privacy. http://nomi.com/privacy/.

[14] Radius Networks Privacy Policy. http://www.radiusnetworks.com/privacy-policy.html.

[15] ReadMe systems - Privacy. http://readmesys.net/privacy.html.

[16] RetailNext - Privacy Policy. http://retailnext.net/policies/privacy-policy/.

[17] Solomo - Privacy Policy. http://www.solomotechnology.com/privacy-policy.html.

[18] Turnstyle Analytics INC. Privacy Policy. http://www.getturnstyle.com/tos/privacy.html.

[19] Walkbase - Privacy. http://www.walkbase.com/optout.

[20] IEEE Standard for Information technology–Telecommunications and information exchange between systems local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, 2012.

[21] Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *IEEE Symposium on Security and Privacy, S&P 2012*, pages 538–552, San Francisco, CA, USA, May 2012. IEEE Computer Society.

[22] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11(0):56 – 69, 2014.

[23] Anthony C Davies, Jia Hong Yin, and Sergio A Velastin. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, 7(1):37–47, 1995.

[24] Solar Designer. John the Ripper password cracker, 2014. `http://www.openwall.com/john/`.

[25] Ben Greenstein, Ramakrishna Gummadi, Jeffrey Pang, Mike Y. Chen, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Can Ferris Bueller still have his day off? protecting privacy in the wireless era. In *Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, pages 10:1–10:6, Berkeley, CA, USA, 2007. USENIX Association.

[26] P. Leach, M. Mealling, and R. Salz. RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace, 2005. `http://www.ietf.org/rfc/rfc4122.txt`.

[27] J.L. Massey. Guessing and entropy. In *International Symposium on Information Theory, ISIT 1994*, pages 204–, Trondheim, Norway, June 1994. IEEE.

[28] 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.

[29] Future of Privacy Forum. Mobile Location Analytics Code of Conduct, 2013. `http://www.futureofprivacy.org/wp-content/uploads/10.22.13-FINAL-MLA-Code.pdf`.

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

[31] 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.

[32] Jens Steube. hashcat advanced password recovery, 2014. `http://hashcat.net`.

Contents

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399