RoMA: Rotating MAC Address for privacy protection

Johann Hugon, Mathieu Cunche, Thomas Begin
University of Lyon, INSA-Lyon, Inria, UCB Lyon 1, ENS Lyon, CNRS, CITI Lab, LIP Lab.
Lyon, France

ABSTRACT

MAC addresses can be collected by passive observers to track Wi-Fi users. While address randomization neutralizes this threat for devices not yet associated, the problem remains for devices being associated with a WLAN. In this paper, we introduce RoMA, which is an anti-tracking scheme making use of concurrent VIFs. We provide a proof-of-concept implementation of RoMA and show that it has a limited impact on the performance of the devices.

1 INTRODUCTION

Wi-Fi has become ubiquitous in our everyday life, with more than 18 billion devices (e.g., smartphones, tablets, laptops ...) embedding this technology in 2022 [1]. On the flip side, Wi-Fi can be leveraged by passive observers to track users in the physical world through the collection of MAC addresses included in wireless frames [8]. To prevent this obvious privacy threat, the concept of address randomization[4] has progressively been adopted by vendors [7, 9] and is being integrated in networking standards [10]. With address randomization, a device does not include its real MAC address in the frame header, but rather uses a randomly generated address that is changed periodically.

Address randomization works seamlessly when the device is not yet connected to a WLAN. Indeed, changing the MAC address does affect the discovery mechanisms and their probe requests. However, when the device is associated with a WLAN, address randomization would inevitably result in a full re-association procedure between the device and the WLAN and therefore a loss of connectivity of several seconds.

In this paper, we introduce a solution to enable the efficient use of address randomization for associated devices. Our approach relies on the use of a pool of rotating virtual interfaces. Address randomization is carried out by periodically renewing virtual interfaces; persistent connectivity for the device is obtained by ensuring that, at any time, at least one virtual interface is connected to the WLAN. Our contributions are as follows: We introduce the concept of virtual interface as an anti-tracking solution in WLANs; We present the design of RoMA, an anti-tracking scheme based on virtual interface rotation; We present a functional proof-of-concept of our scheme; We present an experimental evaluation demonstrating the small impact of RoMA's on performance.

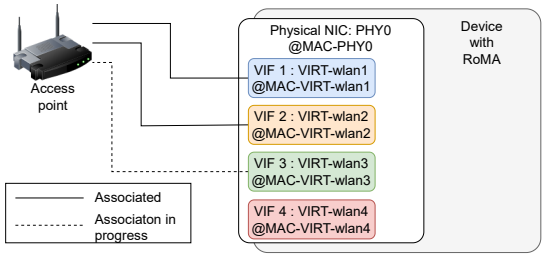

Figure 1: Architecture of RoMA: A physical interface supports a pool of rotating virtual interfaces to protect against tracking.

2 DESCRIPTION OF ROMA

To hinder the tracking of end-user devices through their MAC address while they are associated to an Access Point (AP), we propose Rotating MAC Address (RoMA). RoMA strengthens the privacy of end-users using a pool of MAC addresses on their device. The constituents of these pools are regularly renewed to blur the bond between a MAC address and an end-user device.

The implementation of RoMA raises two main challenges. First, to maintain a continuous connectivity, we need a way to have multiple concurrent interfaces whereas most end-user devices are equipped with a single physical network interface. RoMA makes use of virtual network interfaces (VIFs) to address this first issue. Second, we need the OS of the end-user devices to seamlessly dispatch the traffic among the multiple VIFs (be they connected to the same AP or to different APs of the same WLAN). To this aim, RoMA implements a dispatcher that relies on kernel functions of the `iproute2` Linux package to efficiently handle the resulting routing and load balancing issues. Figure 1 illustrates a high-level description of RoMA. We now describe in more detail how RoMA works.

Handling a pool of concurrent VIFs

A VIF is an abstract representation of a computer network interface that corresponds to a physical network interface, possibly shared with other VIFs. VIFs were introduced with Linux 5.04 and are supported by all recent Linux distributions and Android versions. In practice, they are mostly used to connect a device to multiple networks or to allow a device to act both as a station and an AP. An end-user device may instantiate multiple concurrent VIFs, each having its own distinct MAC address and IP address (assigned by the DHCP

server). One can use the *iw* command to create and destroy a VIF. In the case of RoMA, we generate and maintain a pool of L VIFs, all linked to the same physical network interface. We use *macchanger* to set the MAC address of the new VIFs. Various policies are possible for the renewal of the pool of VIFs. The lifespan of a VIF may be based on its activity (e.g., expressed as a maximum number of connections or packets) or on its lifetime (e.g., expressed as constant or random time). In our experiments, the lifespans of VIFs are randomly drawn from a uniform distribution $[0, T]$.

Dispatching the traffic over the VIFs' pool

Having a dynamic pool of virtual interfaces, we need to ensure that the traffic is correctly routed through them. In particular we want to achieve the following properties : 1) traffic should be seamlessly routed through this pool of VIFs and 2) we should be able to control on which VIFs each connection will be routed.

Let us first recall how Linux and Android work when they have multiple possible routes with the same priority for the same destination (which is the case here since all VIFs of the same device have their own default route). The Linux OS relies on ECMP [6] and the hashing function *siphash()* [2], which operates on the classical 5-tuple (made of IP addresses, port numbers, and protocol in use) found in the headers of packets, to select the route. Note that if the routes are of equal weight (no preference), then *siphash()* is devised to ensure a homogeneous distribution over the existing routes. Once the route has been selected, the OS proceeds with the socket creation for the forthcoming traffic. RoMA modifies the way the route and its associated VIF are selected. The current version of RoMA changes the arguments used by the hashing function siphash(). Instead of the classical 5-tuple, RoMA uses the PID (Process Identification) of the process that triggered the request for the socket in order to select the route. By that means, RoMA ensures that all sockets issued by the same PID are associated with the same route, and hence bonded to the same VIF. When a VIF expires, RoMA closes its associated sockets and new sockets are automatically open on different VIFs (selected through the siphash() function) to resume the communications. Controlling the dispatching of the traffic among the VIFs will enable us to thwart certain fingerprinting attacks [5] run by malicious observer. Other strategies for dispatching are possible. For instance, one can include rules in the dispatcher to specify that certain sockets must, or must not, be routed on the same VIF (pairing and exclusion of flows).

3 EXPERIMENTS AND DISCUSSION

We implemented and evaluated the performance of RoMA. Our experimental setup comprises one raspberry and a laptop, acting as the AP and the client, respectively. The client

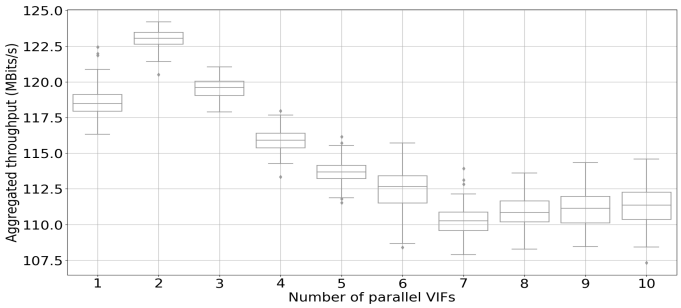

Figure 2: Maximum aggregate throughput when varying the number of VIFs.

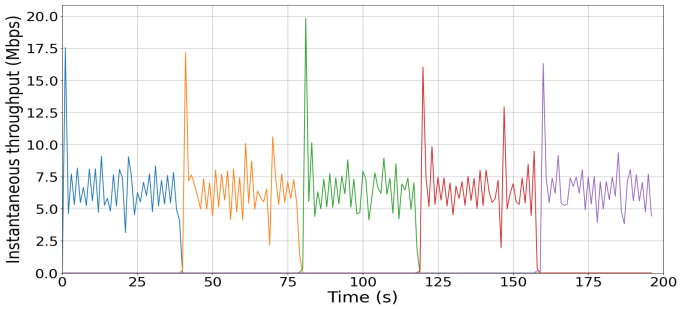

Figure 3: Evolution of the throughput with RoMA changing 4 times the VIF in use to stream a video.

runs Linux 5.18, with dhcpcd and a QUALCOM Artheros AR9462 Wi-Fi card using driver ATH9K. We deployed RoMA on the client. First, we were able to activate up to 10 VIFs on the client. For each number of VIFs, we measure the maximum throughput that can be supported over the whole set of active VIFs. Figure 2 shows that the aggregated throughput peaks at a number of 2 VIFs (due to the overlapping of backoffs [3]) before slightly decreasing (due to the burden of managing several VIFs). Overall, the fluctuations are small, no more than 5%. Our second experiment shows the ability of RoMA to dynamically change the VIFs and thus the associated MAC address for a given connection. To do that, we play a streaming video recorded at 1080p and 60fps from Twitch. Every 40 seconds, RoMA switches the stream to a new VIF. Figure 3 shows the instantaneous throughput attained by the client over time. The impact of VIF switching takes shape in the sudden drop-offs occurring whenever a new VIF takes over. Although the VIF is already up, associated with the AP, (viz WPA2 PSK and IP address through DHCP), RoMA cannot avoid the lag of TCP in increasing its congestion window. However, in our experiments, thanks to the buffering of video frames, these fluctuations did not result in any QoE degradation from the standpoint of the viewer.

Our future works will pertain to the design of advanced strategies to dispatch the traffic over the pool of VIFs.

REFERENCES

[1] Wi-Fi Alliance. 2022. 2022 Wi-Fi trends. (2022). https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-2022-wi-fi-trends

[2] Jean-Philippe Aumasson and Daniel J. Bernstein. 2012. SipHash: A Fast Short-Input PRF. In *Progress in Cryptology - INDOCRYPT 2012*, Steven Galbraith and Mridul Nandi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 489–508.

[3] Andrzej Duda. 2008. Understanding the Performance of 802.11 Networks. In *PIMRC, Proceedings IEEE*, Vol. 8. 2008–1.

[4] Marco Gruteser and Dirk Grunwald. 2005. Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis. *Mobile Networks and Applications* 10, 3 (June 2005), 315–325. https://doi.org/10.1007/s11036-005-6425-1

[5] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. 2020. Protecting against Website Fingerprinting with Multihoming. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 89–110. https://doi.org/10.2478/popets-2020-0019

[6] Christian Hopps. 2000. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992. (Nov. 2000). https://doi.org/10.17487/RFC2992

[7] Lee Hutchinson. 2014. iOS 8 to stymie trackers and marketers with MAC address randomization. (June 2014). http://arstechnica.com/apple/2014/06/ios8-to-stymie-trackers-and-marketers-with-mac-address-randomization/

[8] Transport For London. [n. d.]. Wi-Fi data collection. ([n. d.]). https://www.tfl.gov.uk/corporate/privacy-and-cookies/wi-fi-data-collection

[9] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. 2016. Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. 413–424. https://doi.org/10.1145/2897845.2897883 Core A.

[10] IEEE 802 WG. 2020. IEEE 802E-2020 - IEEE Recommended Practice for Privacy Considerations for IEEE 802(R) Technologies. (2020). https://standards.ieee.org/standard/802E-2020.html