EAI Endorsed Transactions
on Ambient Systems

Research Article EAI.EU

WPA Totem: Sharing temporary Wi-Fi Configuration Tokens using NFC.

Lorenzo Bordoni[1], Armir Bujari[1,*], Claudio E. Palazzi[1]

[1]Department of Mathematics, University of Padua, Via Trieste 63, 35121, Padua, Italy

Abstract

Joining a private Wi-Fi network is an intricate operation for end users, especially when they must type a long and complicated password in order to gain access. Moreover, it is often the case where users connect to multiple such hotspots, placed in public places visited in their daily routine, further exacerbating the issue.

In this paper we present WPA-Totem: a small device that emits Wi-Fi configuration tokens, which can be received by any NFC-enabled smartphones and tablets. Moreover, our proposal is able to automatically change the WPA passphrase after a specified period of time, keeping away the freeloaders from the network.

WPA-Totem is affordable, easy to deploy and compatible with all the existing WPA2-Personal protected Wi-Fi networks. A new client can connect to the access point by simply tapping the totem and this, as we measured, leads to a dramatic improvement of the user experience.

Received on 20 January 2017; accepted on 12 May 2017; published on 28 August 2017

Keywords: Smart Authentication, Wireless Technology, Wi-Fi, NFC

doi:10.4108/eai.28-8-2017.153046

1. Introduction

Mobile device usage has grown rapidly and, according to the 2016 U.S. Mobile App Report [1], they are the leading digital platform, with a total activity accounting for 67% of the time spent on digital media. Their apps alone now represent the 58% of digital media time. On the same lines, the Cisco Visual Networking Index (VNI) forecasts an increase of mobile multimedia traffic in the near future [2].

In this context, most of the mobile applications require access to the Internet in order to fulfill their tasks. To this end, a lot of effort is being devoted in order to provide end users with ubiquitous Internet connectivity through the deployment of next-generation networks [3, 4]. However, current network coverage is not available everywhere, anytime yet and mobile data plans, especially in scenarios of roaming between carriers, are expensive in most countries.

To enhance their hospitality, an increasing number of local administrations and business owners offer to their guests free Internet access inside a private Wi-Fi network, reaching beyond infrastructure coverage [5, 6]. Nevertheless, the network is not entirely private because the passphrase that protects it hardly changes over time and it is passed down orally, or written in signs and menus in shops, pubs and restaurants. In this way, Internet access can be exploited by non returning customers nearby, who have visited the place only once. This *modus operandi* has several implications ranging from degraded quality of service (QoS) for the other users to security concerns in general.

Addressing the issue, in this paper we present an unintrusive, light-weight solution to this problem: WPA-Totem, an active device that is able to share temporary Wi-Fi configuration tokens by exploiting Near Field Communication (NFC) technology. Our solution does not require any intervention at the Internet gateway and is compatible with any commercial of the shelf (COTS) Wi-Fi router. We designed a prototype device which we present in the following along with a preliminary evaluation study showing that the proposal does

*Corresponding author. Email: abujari@math.unipd.it

fulfill its design objective, improving user experience while guaranteeing a higher level of security of public networks.

The rest of the article is organized as follows: in Section 2 we provide a concise background overview on the technologies involved, necessary for understanding the rest of the paper. Next, in Section 3 we outline some key requirements for a successful product and in Section 4 we detail the components of our prototype. Quantifying the usability improvement, in Section 5 we present and discuss the results of an experiment involving fifteen users from the University of Padua, Italy. In Section 6 we compare our work to potential similar products and then in Section 7 we identify new directions that can make WPA-Totem a definitive solution. Finally, in Section 8, conclusions are drawn.

2. Background

In order to fully understand how WPA-Totem works, three main concepts should be demystified: Wi-Fi Protected Access (WPA) modes, NFC technology and NFC Data Exchange Format (NDEF).

2.1. Wi-Fi Protected Access (WPA)

After researchers found several weaknesses in the key scheduling algorithm of RC4 [7, 8], a key technology used in the Wired Equivalent Privacy security protocol (WEP, part of the IEEE 802.11 standard [9]), the Wi-Fi Alliance in 2003 proposed the Wi-Fi Protected Access protocol (WPA, referred to as the draft IEEE 802.11i standard [11]) to promptly mitigate the flaw, pending the availability of the full IEEE 802.11i standard [12], which was later published in 2004, specifying a more secure WPA2 protocol. The common goal of these protocols is to provide confidentiality over a wireless network.

According to how the encryption keys are distributed, different WPA versions have been proposed, i.e., WPA-Personal and WPA-Enterprise. WPA-Personal, also referred to as WPA-PSK (Pre-Shared Key) mode, is more suitable for home and small office networks, because it does not require the deployment of a centralized authentication server. Each client encrypts the wireless network traffic using a secret key, which is set on the access point as a string of 64 hexadecimal digits, or as a passphrase of 8 to 63 printable ASCII characters, and must be entered by users when connecting to the network. WPA-Enterprise (also referred to as WPA-802.1X mode) is designed for business environments and requires the deployment of a RADIUS server handling user authentication. In this case, the encryption keys are securely created and assigned on a per session basis, after the user presents the login credentials.

2.2. Near Field Communication

While radio frequency identification (RFID) is designed primarily to identify and track objects by interrogating RFIC bundled tags, NFC is a short-range wireless technology that expands RFID capabilities and enables two-way connectivity between devices [13]. It operates within the unlicensed radio frequency (ISM band of 13.56 MHz) at rates ranging from 106 to 424 kbps. The interaction between two NFC-enabled devices is contact-less and requires a distance of 4 centimeters or less. Three main modes of operation are typically employed:

- Reader/writer mode, where a NFC device is used to read or write passive tags, like in RFID.
- P2P mode, where a NFC device is used to share data with other peers.
- Card emulation mode, where a NFC device itself acts as a tag, which can then be read by another NFC device.

Depending on the scenario and deployment strategy, one mode is best suited instead of the other. For more information regarding this topic we refer the reader to [13]. We anticipate that our solution operates in the P2P mode and information exchange occurs by simply tapping the devices.

2.3. NFC Data Exchange Format (NDEF)

NFC Data Exchange Format (NDEF) specifies the common language and format used to exchange information between NFC devices and tags. It is a binary format comprised of multiple records. At a basic level, an NDEF record is composed of an application-specific payload (i.e., the actual content) and an header that describes its type and length. Transactions are usually short and, when more throughput is required, a longer term communication through other channels (e.g., Wi-Fi or Bluetooth) is instantiated. NFC and NDEF standards are defined and maintained by the NFC Forum.

The Android operating system by Google, from version 5.0 Lollipop, is able to read/write WPA-Personal configuration tokens from/to NFC tags on NFC-enabled mobile devices. The *Set up Wi-Fi NFC Tag* dialog is hidden in the *Wi-Fi* menu settings. The Wi-Fi network can be joined by simply tapping the previously written tag and this feature is the cornerstone of our work. Its is noteworthy to point out that in 2014 Apple added a NFC controller to the iPhone 6 and iPhone 6 Plus in order to support the Apple Pay mobile payment system. However, at the time of this writing (iOS 10 beta), the NFC API is not yet open to third party developers. Furthermore, Microsoft Windows Phones

do not have this feature in their operating system. For these reasons, we chose to tailor our solution to the Android platform; however, through Section 7, we sketch possible extensions of our approach in order to encompass other platforms as well.

3. Toward an Unintrusive and Light-weight Solution

We designed WPA-Totem with local businesses and public places in mind. In order to make the product appealing and cost-effective while addressing the stated concerns, we defined a list of key requirements which the product must adhere to.

1. It shall be compatible with commercial of the shelf wireless routers and access points and it shall not require significant modification to the existing network infrastructure. The objective should be a plug and play solution.
2. Addressing the freeloader issue, the solution should be able to generate a new password at time intervals specified by the owner, ensuring that the customer is in place when connecting to the network.
3. End users shall not install any application in oder to join the network. Besides being intrusive, this *modus operandi* demands the application being available *locally* as opposed to the marketplace where an Internet connection is required in the first place.
4. The solution shall be easy to configure and, at the same time, it shall be accessible to end users with limited skills. In other words, our solution should not be a deterrent in accessing the network. Keeping the product affordable, it shall be build using inexpensive and easy to find hardware. If this is not the case, the owner of the network would still prefer to write the passwords on paper.

In the next section we discuss the approach taken to meet the listed requirements, making WPA totem an unintrusive, light-weight solution.

4. Meeting the Requirements

In order to meet the first requirement, we planned the architecture shown in Figure 1.

The proposed solution does not require any additional hardware placed at the access point, nor does it require any custom firmware upgrade. For a hassle-free configuration, the setup can be completed through a simple web interface. The only settings that must be adjusted prior to system bootstrap are the following:

- The wireless access point's profile.
- The wireless access point's IP address.

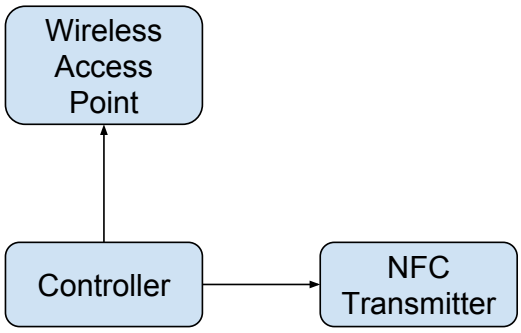

Figure 1. Overall system architecture.

- The wireless access point's username.
- The wireless access point's password.
- The wireless network's service set identifier (SSID).

We define a wireless access point's profile as a community-contributed file that contains information on how to change the WPA settings on a specific model.

The WPA-Totem software in implemented in Python, using the Flask microframework which serves the web interface and the *nfcpy* module dealing with NFC data exchange. After a specified time interval value, parameter configured through the web interface, the software updates through an HTTP(s) request the WPA passphrase of the wireless gateway. The passphrase is randomly generated and is comprised of mixed-case alphanumeric, 63 characters long. This also corresponds to the maximum allowed passphrase length. This feature, also allows us to meet the second requirement. Successively, a well-formed Wi-Fi configuration token is then transferred to a mobile device, through a simple touch of the WPA-Totems' surface. This is achieved via NFC exploiting the Android Beam feature, a P2P-like operational mode.

In order to meet the third requirement we must exploit the operating system's capabilities; however, there is a lack of documentation on how to connect to a Wi-Fi network via NFC. In specifics, the data format used to exchange the configuration is not described in the Android's documentation. Addressing this issue, we made some effort to reverse engineer the operating system's source code and some tags written with the *Set up Wi-Fi NFC Tag* feature. We dumped a NFC tag, which was previously configured with *WPATotemDemo* SSID and *SecretPassphrase* passphrase, which resulted in the following raw data:

```
17d2 614d 7070 696c 6163 6974 6e6f 762f
646e 772e 6166 772e 6373 0e10 3f00 2610
0100 1001 0045 570c 4150 6f54 6574 446d
6d65 106f 0003 0002 1020 000f 0002 1008
0027 5310 6365 6572 5074 7361 7073 7268
```

```
7361 1065 0020 0006 0000 0000 1000 0049
0006 2a37 0100 0020
```

It is a valid NDEF message with a single short record, as shown in Table 1 and described in the NFC Data Exchange Format (NDEF) Technical Specification [14], where, in this case, MB and ME equals to 1, CF and IL equals to 0. TNF is set to 0x02, which corresponds to a media-type as defined in RFC 2046, and TYPE is *application/vnd.wfa.wsc* (TYPE LENGTH equals to 23). Since the IL flag is not set, the ID LENGTH and ID fields are omitted from the record. Table 2 summarizes the first part of record. The actual configuration is contained in the payload and its structure can be derived from the *NfcWifiProtectedSetup* class in the Android's source code.

7	6	5	4	3	2	1	0
MB	ME	CF	1	IL	TNF		
TYPE LENGTH							
PAYLOAD LENGTH							
ID LENGTH							
TYPE							
ID							
PAYLOAD							

Table 1. NDEF Short-Record Layout (SR=1)

7	6	5	4	3	2	1	0
1	1	1	1	1	0x02		
23							
PAYLOAD LENGTH							
application/vnd.wfa.wsc							
PAYLOAD							

Table 2. Actual NDEF Short-Record

WPA-Totem supports the WPA2-Personal profile. Although it would have been trivial to extend the support to WPA-Personal, research studies have pointed out some vulnerabilities [15]; hence, we preferred to force the adoption of the more secure WPA2. WPA-Enterprise was not taken into consideration since it would require a complex server-side configuration, colliding with our fourth requirement.

To keep the cost of production under $100 and meet the last requirement, we assembled our prototype using the components described in Figure 2.

The main controller is a Raspberry Pi, a low-cost ($20-$35) credit-card sized Linux-based development board. We chose the latest Raspberry Pi model (third generation, up to the moment of this writing), but previous versions and other embedded computers can

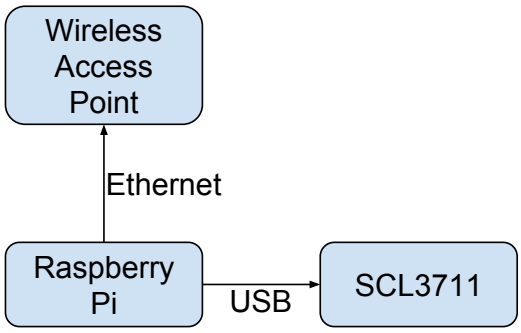

Figure 2. Prototype components

be used, as long as they can run on Linux and they are equipped with an ethernet and USB port.

As NFC transceiver we adopted the SCL3711 (approximately $50) by Identive Infrastructure (formerly SCM Microsystems), which integrates the NXP PN533 NFC module. When connected to a USB 2.0 port, in conjunction with *nfcpy*, the transceiver is able to read/write NFC Forum Type 1/2/3/4 Tags, emulate Type 3 Tags and act as a Peer2Peer Target.

We tested WPA-Totem deliberately with a COTS and discontinued USRobotics USR9113 Wireless Ndx ADSL2+ Gateway at our disposal, to show that it does not require state of the art hardware.

We designed a custom case to fit all the components, which can be placed on a table or where deemed appropriate. Our prototype can be seen in Figure 3a, whereas Figure 3b reveals its internals consisting of the Raspberry Pi (the main board) and SCL3711 (the black USB dongle).

(a) Bundeled product **(b)** Product internals

Figure 3. The final product: in the left is shown the bundled product, while in the right are shown its internals.

5. Evaluation

In order to evaluate the benefits of WPA-Totem on the user experience, we asked fifteen users of different genders, age and skill, to individually connect with their smartphones to our Wi-Fi network which we

previously protected with a mixed-case alphanumeric, 8 characters long passphrase. We explained to the participants the task at hand before starting a stopwatch, that we stopped after they entered the correct password, when they touched *Connect* available through the user interface. In the same way, we explained them the principle behind WPA-Totem and we measured the time required to configure the Wi-Fi network.

Figure 4 shows that using WPA-Totem requires significantly less time when compared to the other *modus operandi* of manually configuring the Wi-Fi network, since it requires only few touches *Connect* of the screen. It is noteworthy to point out that the effort of this interaction in the case of adopting our prototype is independent to the length of the passphrase. While the population sample size used to evaluate our proposal consists of a few participants, the outcome is undoubtedly in favor of WPA-Totem.

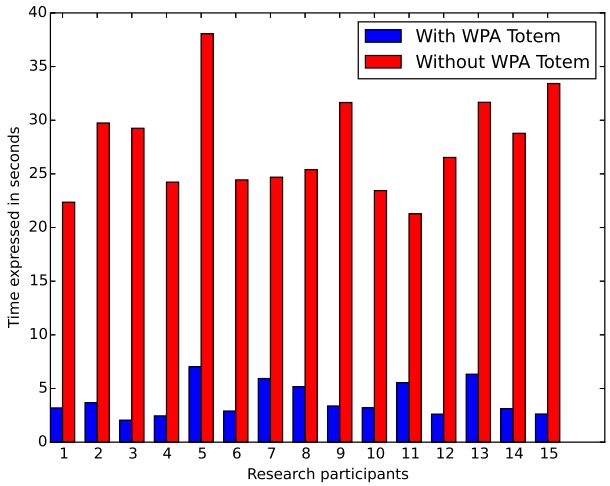

Figure 4. Wi-Fi configuration time with and without WPA-Totem

6. Related Work

NFC has already been exploited as a solution for user account provisioning in public Wi-Fi networks [16], but its adoption is mainly aimed at large scale providers. This approach offers a more secure system by requiring mutual authentication between service providers and users, using EAP-TLS. However, its deployment is excessively complex for small businesses, hence it violates our first and fourth requirements. Also, their implementation goes contrary to our third requirement, because it requires a client-side application handling the configuration. We would like to emphasize that WPA-Totem is designed with small local businesses in minds, where a wide distribution of the Wi-Fi connectivity is usually not needed.

Similarly, other researchers have shown how a NFC side-channel can be exploited, in conjunction with an asymmetric cryptographic system, to address evil twin attacks, captive portal eavesdropping, and man-in-the-middle attacks in public networks [17]. WPA-Totem is developed to improve user experience, usability and ease of deployment, while ensuring the same level of security offered by default by WPA2. It might be possible to integrate WPA-Totem with the above solutions, making it a more robust system, but only by relaxing one or more of our requirements.

To the best of our knowledge, the closest thing to WPA-Totem is the Wi-Fi Protected Setup (WPS, originally Wi-Fi Simple Config). Introduced in 2006 by the Wi-Fi Alliance, it is an attempt to simplify the setup of Wi-Fi Protected Access, allowing to connect a device to the network and enable data encryption by pushing a button (push-button configuration method), through a NFC interface or by entering a unique PIN (PIN entry method), which is the only mandatory method in all Wi-Fi Protected Setup certified devices. Besides requiring a wireless access point that supports WPS, the NFC method is not as powerful as WPA-Totem since it is not able to automatically generate temporary passphrases. Furthermore, please note that researchers proved that brute-force attacks are feasible against a wireless network protected with PIN-based WPS, both online [18] and offline [19].

7. Discussion

As a prototype, WPA-Totem is not feature-rich and several enhancements can be further implemented. At the moment, the system supports NFC-enabled Android mobile devices but it can further be extended in order to support other platforms. This however requires that we relax the third requirement since a software solution is needed in order to decode the Wi-Fi configuration message needed in order to set up the device.

Some devices running the Microsoft Windows OS are equipped with a NFC transceiver, but NDEF data must be translated to the Wi-Fi configuration by a third party application. Differently, on NFC-enabled Apple devices the communication is restricted to the Apple Pay payment system.

To support the devices lacking a NFC transceiver, and Apple ones, we could enhance WPA-Totem with a screen which can be used to display a dynamic QR Code, encoding the current Wi-Fi configuration token. This is a feasible workaround, as we can safely assume that most of the devices are equipped with a camera. However, a third party QR Code reader is required to decode the information.

8. Conclusion

In this article we presented WPA-Totem, an unintrusive, lightweight solution addressing the freeloader problem in publicly available Wi-Fi networks, guaranteeing a higher level of security while at the same time achieving a reduced network access time. We started our work by investigating how the Android operating system parses Wi-Fi configuration tokens received via NFC. We reverse-engineered its source code and previously written NFC tags to reveal their internal structure, which has proved to be a NDEF message composed of a single record.

Our main contribution is a working prototype, a device that emits temporary Wi-Fi configuration tokens using NFC, designed with a set of strict requirements in mind. It is a step forward, compared to technologies like Wi-Fi Protected Setup, since it allows customers to access the Internet on the basis of well-defined time slots. Morever, it does not embody the security vulnerabilities present in the former approach.

WPA-Totem can be easily deployed in public places, shops, public transportation and wherever customers remain for a limited time period. It currently supports NFC-enabled Android smartphones and tables but, with a few gimmicks, it can be extended to support products from other major mobile device vendors as well.

Acknowledgment

This work has been partially funded by the Università degli Studi di Padova through the projects CPDA151221 and CPDR142578. Also, the authors would like to thank Alberto Toniolo, BSc student in Architectural Engineering, for his contribution in the design of the WPA-Totem custom case.

References

[1] comScore Inc. (2016) The 2016 U.S. mobile app report. [Online]. Available: http://www.comscore.com/USMobileAppReport2016

[2] Cisco. *Cisco Visual Networking Mobile Forecast* [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[3] N. Saxena, A. Roy and H. Kim, "Traffic-Aware Cloud RAN: A Key for Green 5G Networks," in *IEEE Journal on Selected Areas in Communications*. April 2016; **34**(4):1010–1021.

[4] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," in *IEEE Access* 2015; **3**:1206–1232.

[5] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno and M. Conti, "Data Offloading Techniques in Cellular Networks: A Survey," in *IEEE Communications Surveys & Tutorials*. 2015; **17**(2):580–603.

[6] J. Lee, Y. Yi, S. Chong and Y. Jin, "Economics of WiFi Offloading: Trading Delay for Cellular Capacity," in *IEEE Transactions on Wireless Communications* March 2014; **13**(3):1540-1554.

[7] S. R. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Proc. of Workshop on Selected Areas in Cryptography*, 2001, pp. 1–24.

[8] A. Klein, "Attacks on the RC4 stream cipher,", in *Designs, Codes Cryptography*. September 2008; **48**(3):269–286.

[9] *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, 1997.

[10] *IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std., 2003.

[11] *IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std., 2003.

[12] *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: Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std. 802.11i, 2004.

[13] NFC Forum. *NFC Technical Specifications* [Online]. Available: http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications

[14] *NFC Data Exchange Format (NDEF) Technical Specification*, NFC Forum Std. NDEF 1.0, 2006.

[15] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proc. of ACM Conference on Wireless Network Security*, ser. WiSec '09. New York, NY, USA: ACM, 2009, pp. 79–86. [Online]. Available: http://doi.acm.org/10.1145/1514274.1514286

[16] A. Matos, D. Romão and P. Trezentos, "Secure Hotspot Authentication through a Near Field Communication Side-channel," in *Proc. of ACM Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2012, pp. 807–814.

[17] Y. Nobu, K. Takeda and H. Yamaki, "Implementation of a User Account Provisioning System based on NFC for Public Wi-Fi Services," in *Proc. of Conference on Computer Application Technologies*, Aug 2015, pp. 114–117.

[18] S. Viehböck. (2011) Brute forcing Wi-Fi Protected Setup. [Online]. Available: https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf

[19] D. Bongard. (2014) Offline bruteforce attack on WiFi Protected Setup. [Online]. Available: https://passwordscon.org/wp-content/uploads/2014/08/Dominique_Bongard.pdf