WI-FAB: Attribute-Based WLAN Access Control, Without Pre-shared Keys and Backend Infrastructures

Claudio Pisa
CNIT, Italy
claudio.pisa@uniroma2.it

Alberto Caponi
CNIT, Italy
alberto.caponi@uniroma2.it

Tooska Dargahi
CNIT, Italy
tooska.dargahi@uniroma2.it

Giuseppe Bianchi
University of Rome Tor Vergata, Italy
giuseppe.bianchi@uniroma2.it

Nicola Blefari-Melazzi
University of Rome Tor Vergata, Italy
blefari@ing.uniroma2.it

ABSTRACT

Two mainstream techniques are traditionally used to authorize access to a WiFi network. Small scale networks usually rely on the offline distribution of a WPA/WPA2 static pre-shared secret key (PSK); security hence relies on the fact that this PSK is not leaked by end user, and is not disclosed via dictionary or brute-force attacks. On the other side, Enterprise and large scale networks typically employ online authorization using an 802.1X-based authentication service leveraging a backend online infrastructure (e.g. Radius servers/proxies). In this work, we propose a new mechanism which does not require neither online operation nor backend access control infrastructure, but which does not force us to rely on a static pre-shared secret key. The idea is very simple, yet effective: directly broadcast in the WLAN beacons an encrypted version of the secret key required to access the WLAN network, so that only the users which possess suitable authorization credentials can decrypt and use it. This proposed approach clearly decouples the management of authorization credentials, issued offline to the authorized end users, from the actual secret key used in the WLAN network, which can thus be in principle changed at each new user's access. The solution described in the paper relies on attribute-based encryption, and is designed to be compatible with WPA2 and deployable within standard 802.11 management frames. Since no user identification is required (access control is based on attributes rather than on the user identity), the proposed approach further improves privacy. We demonstrate the feasibility of the proposed solution via a concrete implementation in Linux-based devices and via relevant testing in a real-world experimental setup.

Keywords

Attribute-Based Access control; Attribute-Based Encryption; WPA; WPA2; WLAN Federation; Privacy Preserving;

HotPOST16, July 04-08, 2016, Paderborn, Germany

DOI: http://dx.doi.org/10.1145/2944789.2949546

1. INTRODUCTION

With the increasing number of smart mobile devices, mobile users are willing to have ubiquitous access to the Internet. As predicted by Cisco [7], the number of personal mobile devices will grow to *8.2 billion* by 2020. However, the existing cellular network (i.e., 3G, 4G, LTE) is not able to support this growing demand of mobile users. As a consequence, the widely available WiFi systems are considered to be the main choice for offloading the data traffic [8, 27]. Cisco [7] predicted the amount of offloaded traffic from 3G and 4G to increase to 48% and 58%, respectively, by 2020. However, open un-protected WLANs, deployed in several public locations, are vulnerable against security attacks [24], for which several protocols have been designed in order to tackle this issue [14].

An important challenge in using WiFi connections is to provide a secure and convenient way for user authentication and access control. Typically, upon connecting to an open WiFi network, the user is presented with a splash page that requires to authenticate or register. On the other side, access control in protected WLANs is non-trivial: the user needs credentials, which have to be obtained through another channel or offline, to connect to a protected network. Although traditional security protocols, i.e., Wired Equivalent Privacy (WEP) and WiFi Protected Access (WPA) are prone to several security attacks [14, 17], these can be prevented by employing WPA2, with e.g. IEEE 802.1X [6]. However, when a service provider deploys such advanced authentication mechanisms, or even Radius-based authentication federations, it is required to setup and maintain online, interactive authentication infrastructures. In this scenario, an untrusted WiFi Access Point (AP) might threaten users' privacy, since a curious service provider would be able to track the clients connected to the APs [16].

We believe that attribute-based access control (ABAC) [22] is a promising solution for providing secure, privacy-preserving authentication and access control in such scenarios. ABAC is an access control mechanism in which the access of the users to a specific content/resource/object is specified based on the attributes of the user (e.g., occupation). The main advantage of ABAC compared to the other access control mechanisms, such as role-based or identity-based access control, is its flexibility especially in dynamic access control decisions, where there is no a priori information about the users, and large scale scenarios [21].

Recently, several researchers tend to adopt Attribute-Based

Encryption (ABE) [25] to provide privacy and ABAC solutions in several scenarios, such as online social networks [12], cloud computing [23], and location-based services [18]. ABE provides fine-grained access control over data through defining attribute-based access policies. In particular, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [13], which is an instantiation of ABE, allows the data owner to encrypt data specifying expressive access control based on a set of attributes. Only the users who have the right attributes in their decryption keys, will be able to decrypt the ciphertext.

In this paper, adopting CP-ABE, we propose *WI-FAB*, an Attribute-Based WLAN access control mechanism, without pre-shared keys and backend infrastructures. In our proposed approach, we introduce a clear separation between the authentication and issuance infrastructure, and the Authorization infrastructure. In particular, we encrypt the WPA2 secret, utilized to secure the WiFi connection, using CP-ABE and then divide it into several chunks. We then insert each chunk in the WLAN beacons and broadcast it in the network. Only the users who can rebuild the information included in beacons and decrypt it, and hence can retreive the WPA2 secret, are authorized to connect to the network. To the best of our knowledge, the proposed approach is the first that does not require any pre-shared key. Through extensive experimental results, we show that WI-FAB is secure, efficient and scalable.

2. BACKGROUND

In this section we provide background knowledge on the concepts that we adopt in our proposed approach.

Attribute-Based Encryption

Attribute-Based Encryption (ABE) [25], is a powerful public key encryption scheme, in which encryption and decryption are based on descriptive attributes (such as age, gender, or occupation). Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [13] and Key-Policy Attribute-Based Encryption (KP-ABE) [20] are the two main types of ABE. In CP-ABE, the data owner enforces an access policy on the ciphertext. A user will be able to decrypt the ciphertext, if and only if, her decryption private key satisfies the defined access policy. While, in KP-ABE the access policy is bound to the decryption key of the user. She is able to decrypt a ciphertext if the attributes specified on the ciphertext matches her key's access policy. Since CP-ABE provides the data owner with a means to have more control over the data, more researchers have concentrated on adopting CP-ABE in several applications [26]. In the following we provide explanation of the CP-ABE basic functions:

- **Setup.** Taking a security parameter as input, it outputs the public parameter PK, and a master key MSK.
- **KeyGen.** Taking a set of attributes S_U, the master key MSK and the public parameter PK, it outputs a decryption key SK_{S_U} reflecting the given attributes.
- **Encryption.** Taking as input a message M, an access policy Π, and the public parameter PK, it outputs the ciphertext E.
- **Decryption.** Taking as input the ciphertext E that is encrypted under the access policy Π, the decryption key SK_{S_U}, and the public parameter PK, it outputs the message M if and only if S_U "satisfies" the access policy Π.

ABE has several advantages compared to the other pubic-key encryption methods [10]: (i) ABE provides fine-grained access control over data through allowing the data owner to define expressive access policies based on the attributes; (ii) the proposed approaches based on ABE are scalable and independent of the number of authorized users; (iii) ABE is efficient in terms of communication, storage and key management overhead. This is due to the fact that there is no need for sharing any secret between the parties.

Attribute-Based Access Control

Attribute-based access control (ABAC) [22, 19] is a flexible access control method in which the acceptance or rejection decision for accessing a resource is made based on the attributes of the requester. ABAC is indeed efficient in terms of communication overhead between the requester and the resource owner. This is due to the fact that the two parties do not need to agree on a pre-shared key to access the resource. Moreover, ABAC preserves the privacy of the users in the sense that the access credentials are not bound to the user identity. Instead, the resource owner only defines the access policies for the resource, and the user will be authorized to access the resource if and only if her credentials, which are bound to her attributes (such as citizenship or group membership) satisfy the access policy.

WPA2 Protocol

The Wi-Fi Protected Access 2 (WPA2) protocol [9] is a rectification of the 802.11 standard, which is introduced in order to address the security vulnerabilities of the Wi-Fi Protected Access (WPA) protocol for wireless networks. WPA2 supports the use of Advanced Encryption Standard (AES) in order to provide data confidentiality and integrity. Moreover, WPA2 provides both personal and Enterprise authentication capabilities [11]: in the personal authentication method, WPA2 makes use of Pre-Shared Key (PSK), while, in the Enterprise mode, the users need to be authenticated based on the IEEE 802.1X.

IEEE 802.11 Beacon Management frames

The IEEE 802.11 standard [5] defines several subtypes of management frames. Among these, *Beacon* frames are broadcast periodically by the access point to advertise its presence, provide the SSID (i.e. the name of the wireless network) and announce its capabilities and other parameters to other wireless devices within its range. These data included in Beacons are enclosed in a sequence of field tuples called *Information Elements*. Of specific interest for this work are *Vendor-Specific* Information Elements, which are used to carry information which is not explicitly defined in the IEEE 802.11 standards.

Fountain Coding

Digital fountain (also known as fountain coding) first introduced in 1998 in order to provide a reliable distribution of bulk data to a large number of users [15]. Fountain coding allows a data owner, who wishes to send a data consisting of a sequence of m equal length packets, to send a stream of distinct packets (called encoding packets or droplets) into the network. The receiver will be able to reconstruct the

source data by receiving any subset of the encoding packets composed of exactly m number of packets. Fountain coding is reliable, in the sense that it guarantees all the intended users will receive the data source. Moreover, it provides an efficient and on-demand method of sending data to the user in a lossy environment.

3. PROPOSED APPROACH

We have devised a privacy-preserving attribute-based access control mechanism for resources which are protected by a shared secret, leveraging the CP-ABE scheme proposed in [13]. Figure 1 shows an overview of our proposed approach. We use CP-ABE to encrypt a secret that grants a single access to a resource.

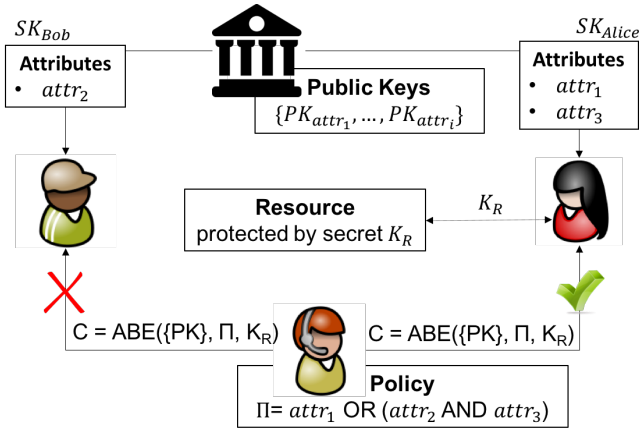

Figure 1: CP-ABE based proposed ABAC mechanism

An authority A generates a master key MSK, a public key for each attribute it wishes to support, and also a key SK_{S_U}, associated to a set of verified and appropriate attributes S_U, for each user U.

Let K_R be the secret that protects a single access to a resource R, and Π the attribute-based policy that a user V holding R wants to enforce on the access to this resource. Then V encrypts K_R using CP-ABE according to the policy Π and publishes the encrypted secret $C = ABE_{\Pi}(K_R)$.

The users which own the keys whose set of associated attributes satisfy the policy Π can successfully decrypt C and obtain K_R. K_R can subsequently be employed to access the resource R once.

3.1 WI-FAB

We employ the above-described access control mechanism in a scenario in which the resource is a wireless network protected by a WPA2 secret. We call the below-described system WI-FAB, outlined in Figure 2. For each user U, an authority A, which can verify a set of attributes S_U of U, releases a private key SK_{S_U}. A WLAN service provider W that trusts the authority A, for each of its access points AP in which it wants to enforce the policy Π:

1. generates a new random secret K_W
2. sets K_W as the WPA2 secret of the AP
3. encrypts K_W using CP-ABE and the policy Π, yielding $C = ABE_{\Pi}(K_W)$
4. sends C to nearby users in the IEEE 802.11 information elements of the beacons
5. when a user successfully connects to AP, the here described procedure starts again from point 1.

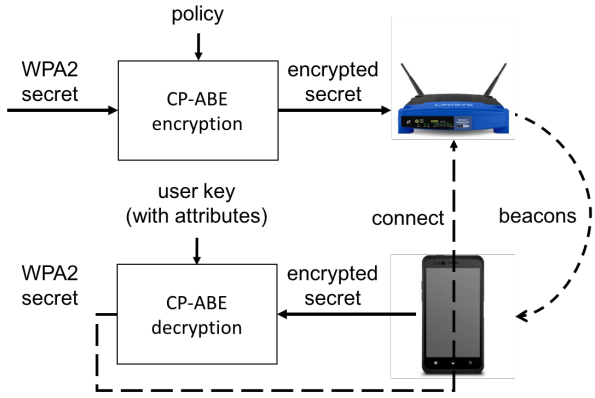

Figure 2: WI-FAB overview diagram

A user U who wishes to access the WLAN service provided by W:

1. captures beacons (by scanning or sniffing) from the access point AP
2. extracts the IEEE 802.11 information elements which contain C
3. attempts to decrypt C using her key SK_{S_U}
4. if the decryption is successful, i.e. if the attributes in S_U satisfy the policy Π, then U holds K_W
5. U generates a random MAC address for her wireless interface
6. U employs K_W to generate a WPA2 configuration and connects to AP using the WPA2 protocol

Note that K_W is used by the WPA2 protocol to generate a per-client session key. Thus, when W generates a new secret for an access point and sets it as the new WPA2 secret, the users which are already connected to that access point are not disconnected. Moreover, the transmission of $C = ABE_{\Pi}(K_W)$ in the IEEE 802.11 information elements of the beacons can be encoded using fountain coding (§ Section 2). In this way if the size of C exceeds the maximum size of a single information element, it can be encoded into smaller droplets. The user can obtain C by capturing enough of these droplets and using fountain (de)coding. Figure 3 sketches the proposed format of the Information Elements contained in the beacons emitted by the access point. The first fields are employed as specified in the IEEE 802.11 standard [5], i.e. the `Element ID` (one octect) contains the value 221, assigned to Vendor-Specific Element IDs; the field `length` (one octect) is set to the aggregated size of the subsequent fields; and `OUI` (three octects) should contain an identifier assigned by IEEE, but for experimentation purposes is temporarily set to an arbitrary (unassigned) value. The `index` field (one octect) is incremented modulo 256 each time that the AP changes the WPA2 secret, and is used by

Stations to discard collected droplets associated to WPA2 secrets which become invalid. The `nchunks` (one octect) and `seed` (two octects) fields contain respectively the total number of chunks in which C has been divided, and the pseudo-random seed used for the current droplet. Finally the `data` field (variable length) contains the actual droplet data.

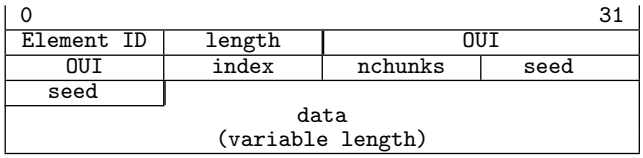

Figure 3: Format of the Vendor Specific Information Elements included in the IEEE 802.11 beacons broadcast by the access point

3.2 Security Analysis

In our proposed system, the WLAN access point makes use of the WPA2 link-level encryption protocol, so that the access to the network requires knowledge of the pre-shared key (PSK). While common deployment of WLAN systems based on WPA2 remains vulnerable against password cracking attacks due to the weakness of the password/passphrase, we propose to generate a random key, of the maximum size allowed by WPA2 at each successful connection so that it cannot be easily guessed, at least assuming the usage of a secure RNG algorithm in the system. However, this challenge-like approach cannot be used without integration of an efficient technology for the distribution of the key that is not pre-shared, i.e., it is a priori unknown to the user. In order to make possible for the users to retrieve the secret key required to access the network by proving that they are authorized, we exploit the CP-ABE scheme. CP-ABE allows us to encrypt the PSK, defining a boolean access policy over the attributes that the user must own in order to be able to decrypt and retrieve the PSK. This strengthen the security of actual WPA2-based systems building an access control mechanism for key distribution on top of it. Assuming honest users that do not leak the CP-ABE secret key to other users with the aim of satisfying the policy, it will be impossible for an attacker to sniff or even forge packets to obtain access to the network without having the credentials required by the encryption policy. Indeed, the random generated WPA2 key will be encrypted by means of the CP-ABE scheme and decrypted only by the legitimate users who are able to satisfy the policy.

4. IMPLEMENTATION

We have performed a Linux-based preliminary implementation of our proposed system, summarized in Figure 4. Although this implementation has not yet been tested on actual embedded and mobile devices, we foresee no big obstacles to its porting to at least other Linux-based operating systems such as OpenWrt/LEDE for the AP part, and Android for the STA part.

Note that our implementation did not require changes to the Linux kernel, but only to userspace tools such as `hostapd` and `iw`, integrated with an implementation of a custom variant of the fountain LT codes and bash scripts. For the CP-ABE part, we are employing the implementation of [13] provided at [1]. Further details are given in the remainder of this section.

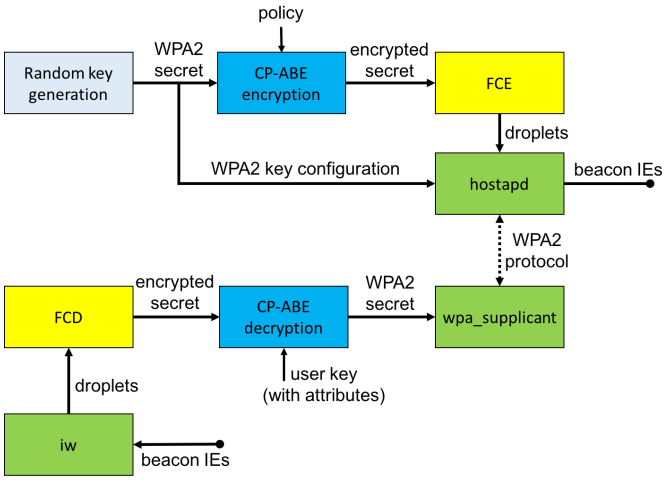

Figure 4: WI-FAB implementation overview

Fountain Coding

We call `FCE` and `FCD` the implementations of the fountain coding encoder and decoder. `FCE` continuously reads the contents of a specific file and provides droplets to a configured named pipe. Droplets include an index number, as described in Section 3.1. When `FCE` is instructed to reload the contents of the file, the index is increased. `FCD` performs the inverse operation: it continuously reads droplets from a named pipe and when is able to reconstruct the original information it writes it to a file.

Access Point

The `hostapd` daemon is a highly configurable user space IEEE 802.11 access point implementation that employs the `nl80211` Linux API. Its source code is publicly available at [2]. Note that `hostapd` already provides a mean to configure static Information Elements to be included in the access point beacons. Our modifications to hostapd allow to: 1) provide a dynamic stream of Information Elements through a local named pipe, and 2) change the configuration of the WPA2-PSK keys without restarting the demon and without disconnecting the users that are already connected. We have devised and implemented a bash script that periodically:

- generates a new random 256 bit WPA2 key S
- encrypts S with CP-ABE using a configurable policy and the supplied CP-ABE public key and stores it in a file F
- configures S as a new WPA2-PSK secret on hostapd
- instructs `FCE` to reload F and to provide its droplets to `hostapd` (through the named pipe).

In order to allow a reasonable time for the connection of the users, we keep the last two generated WPA2-PSKs active at the same time. This is possible as hostapd supports the use of multiple coexistent WPA2-PSKs.

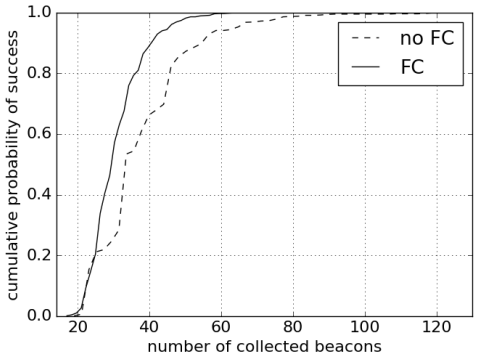

Figure 5: ECDF associated to the number of collected beacons needed to reconstruct the encrypted secret with and without fountain coding (FC)

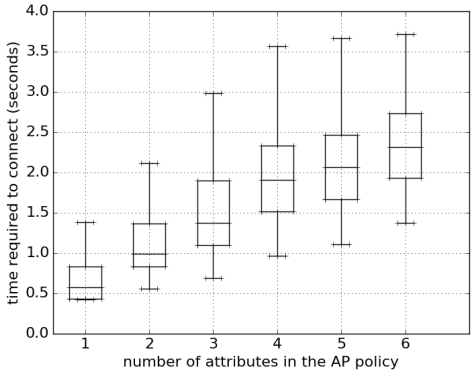

Figure 6: Time needed for the station to connect vs. number of attributes in the AP policy

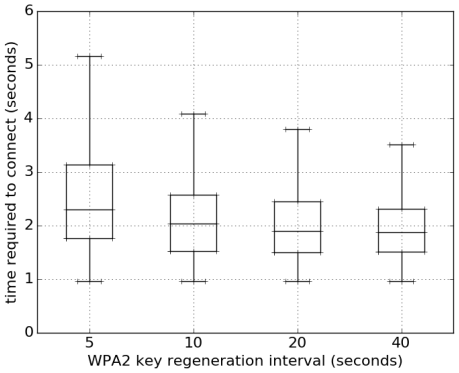

Figure 7: Time needed for the station to connect vs. random WPA2 key regeneration interval (500 connection attempts for each regeneration interval, policy with 4 attributes)

Note that the above implementation is only an approximation of the scheme described in Section 3, as the WPA2 key is changed at regular intervals instead of being changed each time a user successfully connects to the AP.

Station

`iw` is a configuration utility for wireless devices. Its source code is publicly available at [3]. Among its features, it allows to instruct the wireless driver to perform active and passive scans and to output the results. We have performed minor modifications to `iw` to change the way in which the IE contained in the beacons are displayed. In the future we plan to employ `wpa_supplicant` [4] instead, which is a standard tool in Linux-based OS and Android OS, responsible for the connection to IEEE 802.11 WLANs.

We have developed a bash script that runs `iw` based active scans multiple times and sends its output through a named pipe to `FCD`. When `FCD` has collected enough droplets to reconstruct the contents of F, a decryption attempt, using CP-ABE and the configured secret key of the user, is made[1]. If the decryption is successful (i.e. if the supplied secret key of the user is associated to a set of attributes that satisfies the policy configured at the access point), the decrypted WPA2-PSK secret is included in a `wpa_supplicant` configuration file and a connection to the access point is performed, without the need to interact with the user.

5. RESULTS

This section shows the effectiveness of the proposed solution in a real-world experimental setup in our lab. We are employing laptops (on the same desk) with a Linux-based OS running the implementation described in Section 4. The wireless NICs employ the `rt2800usb` driver module. The version employed for the CP-ABE tools is 0.11, while our modified `hostapd` and `iw` are derived from versions 2.5 and 3.3, respectively. Unless explicitly stated, we have configured at the AP a beacon interval of 102.4 ms (i.e. the default value of `hostapd`) and a policy with four attributes.

In the first experiment, we demonstrate the effectiveness of fountain coding as opposed to the approach of just dividing the encrypted secret into numbered chunks. Figure 5 shows the empiric probability (on 1000 samples, i.e. station connection attempts) of recovering the encrypted secret (divided into 16 chunks) vs. the number of collected beacons. Beacon collection make up the majority of the total time needed by the station to successfully connect to the AP. Although our fountain coding implementation has room for substantial improvement, the performance increase of the system is promising.

In the second experiment, we change the size of the policy by varying the number of contained attributes. This changes the size of the encrypted secret and thus the time needed for the station to connect to the AP.

Figure 6 shows how the number of attributes affects the performance, for a series of 500 connection attempts for each set number of attributes. The size of the encrypted secret depends on the number of attributes employed in the policy. In the third experiment we modify the WPA2 key changing interval and observe how this change affects the time needed by the stations to connect to the AP. In Section 3.1 we propose to change the WPA2 secret each time a new station successfully connects to the AP. The results shown in Figure 7 support the feasibility of this proposal.

[1]Please note that this is the only configuration parameter needed at the Station/client side

6. CONCLUSION

We introduced a new attribute-based access control mechanism based on CP-ABE, called WI-FAB. We proposed to employ our mechanism in the scenario of protected WLANs and we implemented a working prototype. We finally showed and discussed the results obtained in a real-world experimental setup.

In the future, we plan to extend the proposed approach to leverage the multi-authority CP-ABE schemes. This will enable federation scenarios in which multiple entities will be able to issue attribute-based credentials to users. Moreover, we plan to investigate further the use of fountain codes, optimizing their usage, the implementation and the employed parameters, as well as other alternative error correction mechanisms.

7. ACKNOWLEDGMENTS

This research was partially supported by the EU Commission within the Horizon 2020 program: ReCRED project grant no 653417 and Bonvoyage project grant no 635867.

8. REFERENCES

[1] Advanced crypto software collection - ciphertext-policy attribute-based encryption. http://hms.isi.jhu.edu/acsc/cpabe/.

[2] Linux wireless - hostapd. http://linuxwireless.org/en/users/Documentation/hostapd/.

[3] Linux wireless - iw. http://linuxwireless.org/en/users/Documentation/iw/.

[4] Linux wireless - wpa_supplicant. http://linuxwireless.org/en/users/Documentation/wpa_supplicant/.

[5] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Mac and phy specifications. *IEEE Std 802.11-2007*, pages 1–1076, June 2007.

[6] IEEE standard for local and metropolitan area networks–port-based network access control. *IEEE Std 802.1X-2010*, pages 1–205, Feb 2010.

[7] Cisco visual networking index: Global mobile data traffic forecast update, 2015 white paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 2016.

[8] A. Aijaz, H. Aghvami, and M. Amani. A survey on mobile data offloading: technical and business perspectives. *Wireless Communications, IEEE*, 20(2):104–112, 2013.

[9] W. Alliance. Wpa2 security now mandatory for wi-fi certified products. *Press Release*, 2006.

[10] M. Ambrosin, M. Conti, and T. Dargahi. On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, pages 49–54. ACM, 2015.

[11] P. Arana. Benefits and vulnerabilities of wi-fi protected access 2 (wpa2). *INFS 612*, pages 1–6, 2006.

[12] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 135–146. ACM, 2009.

[13] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy*, SP'07, pages 321–334. IEEE, 2007.

[14] H. Boland and H. Mousavi. Security issues of the ieee 802.11b wireless lan. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2004.*, volume 1, pages 333–336. IEEE, 2004.

[15] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, 28(4):56–67, 1998.

[16] A. Cassola, E.-O. Blass, and G. Noubir. Authenticating privately over public Wi-Fi hotspots. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS'15, pages 1346–1357. ACM, 2015.

[17] A. Cassola, W. K. Robertson, E. Kirda, and G. Noubir. A practical, targeted, and stealthy attack against wpa enterprise authentication. In *Proceedings of the 20th Annual Network and Distributed System Security Symposium*, NDSS'13, 2013.

[18] T. Dargahi, M. Ambrosin, M. Conti, and N. Asokan. Abaka: A novel attribute-based k-anonymous collaborative solution for lbss. *Computer Communications*, 85:1–13, 2016.

[19] K. Frikken, M. Atallah, and J. Li. Attribute-based access control with hidden policies and hidden credentials. *Computers, IEEE Transactions on*, 55(10):1259–1270, 2006.

[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS'06, pages 89–98. Acm, 2006.

[21] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al. Guide to attribute based access control (abac) definition and considerations. *NIST Special Publication*, 800:162, 2013.

[22] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo. Attribute-based access control. *Computer*, (2):85–88, 2015.

[23] T. Jung, X.-Y. Li, Z. Wan, and M. Wan. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Transactions on Information Forensics and Security*, 10(1):190–199, 2015.

[24] J. S. Park and D. Dicoi. Wlan security: current and future. *IEEE Internet Computing*, 7(5):60, 2003.

[25] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[26] S.-Y. Tan and W.-S. Yap. Cryptanalysis of a cp-abe scheme with policy in normal forms. *Information Processing Letters*, 116(7):492–495, 2016.

[27] H. Zhang, X. Chu, W. Guo, and S. Wang. Coexistence of wi-fi and heterogeneous small cell networks sharing unlicensed spectrum. *Communications Magazine, IEEE*, 53(3):158–164, 2015.