On the Feasibility of Attribute-Based Encryption for WLAN Access Control

Claudio Pisa, Tooska Dargahi, Alberto Caponi, Giuseppe Bianchi, Nicola Blefari-Melazzi
CNIT / University of Rome Tor Vergata, Italy
Email: {name.surname}@uniroma2.it

***Abstract*—User authentication at Wi-Fi Access Points (APs) is becoming an important issue. Wi-Fi APs are indeed ubiquitous, but existing authentication methods such as WPA/WPA2 static pre-shared secret key (PSK), or 802.1X-based online authentication services (e.g., RADIUS servers/proxies) have their theoretical or practical limitations. In a previous work, we proposed WI-FAB, a new authentication mechanism which neither requires online backend access control infrastructure, nor relies on a static pre-shared secret key. In this paper, we extend WI-FAB by removing the need for having a central authority for user authentication and credential issuing. Our main contribution is twofold: (i) adopting *decentralized multi-authority CP-ABE*, we support the users who have authentication/authorization credentials from multiple authorities. We decouple the user credentials issuing from the management of the WPA2-PSK, so that neither the credential issuing authority can track the users, nor the AP can access the real identity of the users. Considering an extensive attack model, we show that the proposed approach is secure and preserves the privacy of the users. (ii) We provide a real-world implementation of the proposed approach on off-the-shelf embedded hardware to demonstrate its feasibility and efficiency.**

***Index Terms*—WLAN Access Control, Attribute-Based Access Control, Multi-Authority Attribute-Based Encryption, Embedded Device.**

I. Introduction

During the last decade, the number of Internet users has increased significantly, which, reported by International Telecommunication Union (ITU), was more than 46% of the world population by the end of 2016 [1]. This is due to the ever increasing number of mobile devices and connections, which will grow to *11.6 billion* by 2021, reported by Cisco [2]. However, due to the limitation of the existing cellular network (i.e., 3G, 4G, LTE), Cisco predicts the increase from 60% in 2016 to 63% by 2021 in offloading from the mobile data onto the fixed network through Wi-Fi or femtocell [2].

At the same time, providing a secure, privacy preserving, and straightforward method for authenticating the users willing to access a Wi-Fi network is challenging. This issue applies to both open and protected Wireless Local Area Networks (WLANs). In the former case, the user usually needs to register on a splash page in real-time, and after being authenticated he will receive the credentials to connect to the Internet. In such a scenario, service providers basically use an 802.1X-based authentication service leveraging a backend online infrastructure. This authentication method not only requires an online infrastructure, which is not always available, but also reveals sensitive information of the users, e.g., their identity or mobility patterns, to the credential issuing authority and the untrusted (or honest-but-curious) access points (APs) [3]. In case of protected WLANs, the user needs credentials, i.e., WPA/WPA2 pre-shared secret keys (PSKs) [4], to connect to the Wi-Fi. These credentials can be obtained offline or through another channel, which introduces several challenges (i.e., how to remember the password or how to keep it safe) and attacks [5], [6].

We believe that, in order to address the privacy and flexibility issues of traditional user authentication methods in WLAN, a desirable solution should satisfy the following requirements: 1) it should be easy for the admin of the network to configure the AP in such a way that the decision on who can access the Wi-Fi can be taken on the fly, by defining real-time access policies; and 2) it should be easy to refresh the WPA2-PSK in order to protect WLANs against unauthorized access (i.e., eliminating the need for updating all the users' credentials). In a previous work, we proposed *WI-FAB* [7], an attribute-based WLAN access control mechanism, without pre-shared keys and backend infrastructures. In WI-FAB, we satisfied the mentioned requirements by encrypting the WPA2-PSK that is used to secure the Wi-Fi connection adopting Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [8]. We enforced an access policy on the encrypted secret key based on the necessary attributes of the users who will be considered authorized to access the Wi-Fi. In WI-FAB [7], we left as future work the scenario in which the defined policy on the encrypted PSK is written over attributes that are related to different authorities/domains. In particular, scenarios in which the users who would like to connect to the Wi-Fi AP are issued credentials from different authorities/domains. There are several real-world applications for such a situation, such as departments of a university that would like to take control of their students/staffs independently, city Wi-Fi, or a building equipped with a central Wi-Fi AP composed of several different independent companies/offices.

In this paper, we adopt *decentralized multi-authority ABE* [9] and extend WI-FAB [7], while we inherit the main advantages of WI-FAB. We encrypt the WPA2-PSK specifying an access policy of attributes from multiple-authorities. We then divide the encrypted PSK into several chunks, insert each chunk in the WLAN beacons and broadcast them in the network. Upon receiving the beacons, a user who wishes to connect to the AP should merge the received chunks and reconstruct the encrypted PSK. If and only if a subset of the

user's attributes, which are associated to his secret key, satisfy the policy associated to the ciphertext, he would be able to decrypt and retrieve the PSK.

Running example: Before introducing the key contributions of the paper, we present a running example, which we will use throughout the paper. Let us consider a university of n departments, e.g., Engineering, Economics, Medicine, History, and so on. Moreover, assume the university campus is equipped with a central Wi-Fi AP. In order to provide Wi-Fi access credentials for the users (i.e., students or staffs) in the campus, in a normal scenario there are two options. First, the users should receive a WPA2 secret from the central authority (CA) of the campus. Second, the campus can use an online backend infrastructure to authenticate the users. In such a scenario, users should receive credentials bound to their identity from the CA of the university. This way, i) users privacy is threatened, i.e., after every access to the Wi-Fi the credential issuing authority can track the users; and ii) there is a need for *online authentication* infrastructure.

This paper provides the following main contributions:

1) Adopting *decentralized multi-authority CP-ABE* [9] we support users having authorization credentials from different authorities;
2) Considering an extensive attack model we show that the proposed approach is secure and preserves the privacy of the users;
3) We provide a real-world implementation of our proposal on consumer-grade embedded hardware and demonstrate its feasibility through experimental results in a real testbed.

II. Background and Related Work

In this section we provide background knowledge on the concepts that we use in our proposed approach, along with a review of the most related work to our proposal in each subsection.

A. Attribute-Based Access Control and Encryption

Attribute-based access control (ABAC) [10], [11] is a flexible access control method in which the acceptance or rejection decision for accessing a data/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. A common tool that is usually used to provide ABAC is *Attribute-Based Encryption*, which we explain below.

1) Attribute-Based Encryption: Attribute-Based Encryption (ABE) [12], is a public key encryption scheme, which enables the data owner to specify fine-grained access policies on the encrypted data. The access policy on the ciphertext is based on descriptive attributes (such as gender, or occupation). Two main forms of ABE are Key-Policy Attribute-Based Encryption (KP-ABE) [13], and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [8]. In the KP-ABE scheme, private keys of the users are associated with the access policies, and a set of attributes is specified on the ciphertext. While in CP-ABE, the access policy is associated to the ciphertext, and private keys of the users are bound to a set of attributes that describes the user. A user is able to decrypt a ciphertext, if a subset of his attributes satisfies the access policy specified on the ciphertext. Due to the characteristics of ABE, several researchers have concentrated on ABE and proposed either new variations of ABE (to mention a few [9], [14], [15]), or new applications and performance evaluation of ABE (to mention a few [16]–[18]).

2) Multi-Authority ABE: One important shortcoming of the first ABE schemes and most of their extensions is that the users' private key issuing is performed by a central trusted authority who should take care of authenticating the users and validating their attributes. Therefore, these schemes are not scalable for several different (distributed) domains in which the users might need to have different credentials reflecting their disjoint attributes (e.g., for their education, occupation, and so on). In order to address this issue, several researchers proposed *multi-authority ABE* to support private keys issued from different authorities having a hierarchical distribution, such as [19], [20]. In 2011, Lewko and Waters introduced a *decentralized multi-authority ABE* [9] scheme in which different authorities do not need to be aware of others, neither rely on a trusted central authority. Hence, there is no global coordination between the authorities other than a setup phase for generation of an initial set of global reference parameters (GP). This enables any party to serve as an authority and issue its public key along with a set of private keys for different users reflecting their attributes.

In our proposed approach, we take advantage of *multi-authority CP-ABE* [9][1] in order to provide more flexibility in attribute selection and access policy definition, as well as removing the necessity of having a central trusted authority. In multi-authority CP-ABE [9] a data owner can define any access policy, composed of any chosen subset of attributes issued by any subset of authorities, over his encrypted data (refer to Figure 1, and for the notations refer to Table I). A user will be able to decrypt the encrypted data if and only if a subset of attributes associated to his private key, issued by any authority, satisfies the access policy on the ciphertext. As depicted in Figure 1, the users U_2 (having attributes from authority A_2) and U_3 (having attributes from both authorities A_1 and A_2) are able to access and decrypt the data. While user U_1 is not able to satisfy the policy and decrypt the data.

In the following, we explain five main algorithms of multi-authority CP-ABE as explained in [9]. Let us consider a set of authorities $\mathbb{A} = \{\mathcal{A}_1, \mathcal{A}_2, ...\}$ each of which having a pair of public key, $PK_{\mathcal{A}_j}$, and secret key, $SK_{\mathcal{A}_j}$, where $j = \{1, 2, ...\}$. Hence, there is a set of public keys, $S_{PK} = \{PK_{\mathcal{A}_j}\}_{j=\{1,2,...\}}$, in the system.

[1]While we are aware of the other more recent multi-authority CP-ABE schemes, such as [21], our main goal here is to show the effectiveness of using multi-authority ABE in anonymous user authentication in WLAN, no matter which of the existing multi-authority ABE schemes is used.

- **Global Setup.** Taking as input a security parameter (λ); it outputs the global parameters GP;
- **Authority Setup.** Taking as input the global parameters (GP); it outputs a pair of public key $PK_{\mathcal{A}_j}$, and secret $SK_{\mathcal{A}_j}$ for each authority $\mathcal{A}_j$. This algorithm is run by each authority, and in practice every authority generates one public/secret key pair for each attribute γ_i that it supports, i.e., it outputs $\{PK_{\mathcal{A}_j,\gamma_i}\}$ and $\{SK_{\mathcal{A}_j,\gamma_i}\}$, while in Figure 1 we simplified this notation by having just one public/secret key pair for each authority;
- **KeyGen.** Taking as input the following parameters: the global parameters (GP), a unique global identity (GID), an attribute (γ_i, where $i = \{1, 2, ...\}$) which belongs to an authority ($\mathcal{A}_j$), and the secret key of that authority ($SK_{\mathcal{A}_j}$); it outputs a key $K_{\gamma_i,GID}$ for a $\langle GID, attribute\rangle$ pair associating the corresponding attribute to the specific identity (i.e., user);
- **Encryption.** Taking as input the following parameters: a message M, an access matrix (A, π), the subset of public keys ($\{S_{PK}\}_{i..j}$) of the relevant subset of authorities ($\{\mathbb{A}\}_{i..j}$), and the global parameters (GP); it outputs a ciphertext CT;
- **Decryption.** Taking as input the ciphertext (CT), the global parameters (GP), and a subset of keys ($\{K_{\gamma_i,GID}\}$) associated to $\langle GID, attribute\rangle$ pair; it outputs the message M if and only if the set of attributes ($\{\gamma_i\}$) associated to the keys "satisfies" the access matrix (A, π) defined on the ciphertext.

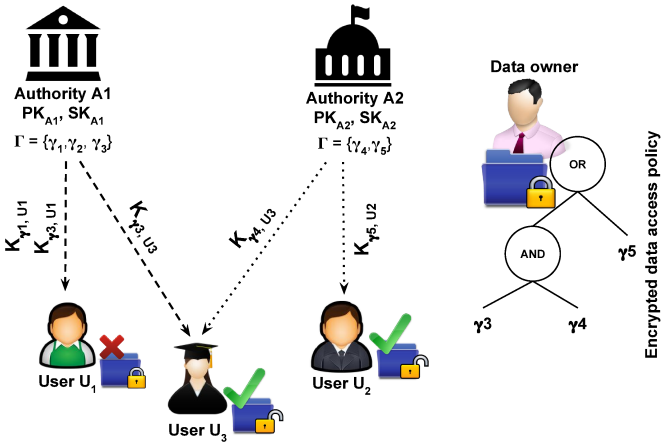

Fig. 1. An example of multi-authority CP-ABE

B. Wi-Fi Authentication and Access Control

The traditional Wi-Fi authentication and access control protocols, i.e., Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) are proved to be vulnerable against several security attacks [5], [6], which was the motivation for the introduction of the Wi-Fi Protected Access 2 (WPA2) protocol [4].

1) WPA2 Protocol: The WPA2 protocol [4] is a rectification of the 802.11 standard that supports the use of Advanced Encryption Standard (AES). WPA2 guarantees data confidentiality and integrity for both personal and Enterprise authentication scenarios [22]. In the Enterprise mode, WPA2 uses IEEE 802.1X [23] in order to authenticate the users, while in the personal authentication scenario, users need to submit a PSK to be authenticated. It should be noted that, after the primary phase of user authentication, WPA2 creates a fresh unique session key for each connected user. This way, the secret key that is used for authenticating the user will be different from the the key that will be used for further communication (i.e., message exchange) encryption.

2) Anonymous User Authentication: Due to the importance of addressing the security and privacy issues of traditional Wi-Fi access control and authentications protocols (as explained in Section I), several methods have been proposed to provide security, while privacy has gained less attention [6]. There are just a few proposals in the literature for providing (anonymous) privacy preserving user authentication. In [6], a private user authentication method based on Private Information Retrieval (PIR) is proposed, which preserves the privacy of the users against the AP and authentication server. Though the proposal is interesting, it still requires an online backend server to provide the users with the credentials for connecting to the AP, which increases user authentication time correspondingly.

Other fields of research related to our context are *anonymous* and *attribute-based* credentials [24], and related systems such as Idemix [25] and U-Prove [26]. However, neither of such schemes are proposed for WLAN access control, nor could be adopted in such a scenario, since they need a two-way communication (generally between a client and a server), while in our considered scenario we have a one-way communication from the AP to the user.

III. Models and Assumptions

In this section we explain our system model and the assumptions that we make in the remainder of the paper, as well as the considered adversary model. We report the notations that we use throughout the paper in Table I.

TABLE I
Notation Table

Notation	Description
GID	User's global identifier
GP	Global parameters
PSK	WPA2 pre-shared secret key
K_s	WPA2 session key
$\mathbb{A} = \{\mathcal{A}_1, \mathcal{A}_2, ...\}$	Set of authorities
$PK_{\mathcal{A}_j}$, $SK_{\mathcal{A}_j}$	Public and secret keys of authority $\mathcal{A}_j$, respectively
$S_{PK} = \{PK_{\mathcal{A}_1}, PK_{\mathcal{A}_2}, ...\}$	Set of public keys of the authorities
$\Gamma = \{\gamma_1, \gamma_2, ...\}$	Universe of attributes
$U = \{U_1, U_2, ...\}$	Set of users
K_{γ_i,U_j}	Secret key of the user U_j ($GID = U_j$) for attribute γ_i
(A, π)	Access matrix
$H()$	Hash function
E_{MABE}	Multi-authority CP-ABE encryption
D_{MABE}	Multi-authority CP-ABE decryption

A. System Model

In our model we consider a network consisting of the following entities: (1) A Wi-Fi access point (AP) which is protected with a WPA2 pre-shared secret key (PSK); (2) A set of users $U = \{U_1, U_2, ...\}$, who intend to connect the Wi-Fi AP; (3) a set of authorities, $\mathbb{A} = \{\mathcal{A}_1, \mathcal{A}_2, ...\}$, who provide the users with a secret key reflecting their attributes.

We assume that each user is holding a unique global identifier, GID, which could be the social security number of the user (consistent with [9], [19]). Each user U_j, for an attribute γ_i, receives a secret key K_{γ_i,U_j} associated to his $\langle GID, attribute \rangle$ pair from the related authority. We also assume that each user can receive secret keys from multiple authorities for several attributes (see Figure 1).

Consistent with [9], in our system any entity (e.g., a building manager, a university or a single department in a university, a city municipality, etc) can become an authority and issue private keys to the users under its domain. We assume that there is no global coordination between these authorities, and they might not even know each other. Each authority can take care of several attributes, and in some special cases, several authorities can take care of the same attribute. Considering our running example (explained in Section I), the *"Student"* attribute is the same in all the departments of a university, and so each department needs to assign secret keys for the *"Student"* attribute to its own students. In order to do so, similar to [9], we consider each attribute to be a string composed of: corresponding authority's public key and the attribute name. For example, for the *engineering* department, e.g., $\mathcal{A}_1$, and *medicine* department, e.g., $\mathcal{A}_2$, the *"Student"* attribute will be $Student.PK_{\mathcal{A}_1}$, and $Student.PK_{\mathcal{A}_2}$, respectively.

We assume that distributed multi-authority CP-ABE algorithms [9] (explained in Section II-A2) are available in the system to be used by the authorities, AP and users.

B. Adversary Model

In this section we explain our considered adversary model, based on which we will provide security analysis in Section IV-A. In this work, we consider two types of attacker: a *passive attacker*, and an *active attacker*. A *passive attacker* can be one of the following entities:

(i) *An external eavesdropper*, who aims at accessing the Wi-Fi network by eavesdropping the channel and trying to obtain the PSK used to protect the Wi-Fi;

(ii) *An honest but curious access point*, who honestly follows the protocol and provides the users with the encrypted version of the PSK, specifying a valid access policy over the attributes; while, it is curious in identifying the users and violating their privacy, e.g., mapping the users' credentials to their identities;

(iii) *An honest but curious credential issuing authority*, who honestly follows the protocol and provides the users with the secret key reflecting their attributes; while, it is curious in violating users privacy, e.g., tracking the users;

(iv) *An internal eavesdropper*, who is a user successfully satisfied the access policy on the PSK and connected to the AP. This attacker aims at violating other users' privacy by eavesdropping the traffic between a connected user and the AP.

An *active attacker* can be one of the following entities:

(i) *A set of colluding users on the attributes*, who do not satisfy the access policy enforced on the ciphertext and try to merge their attributes to obtain the PSK;

(ii) *A set of colluding users on the PSK*, where one user in the set has successfully satisfied the access control on the ciphertext and obtained the PSK. This user aims to share the PSK with unauthorized users. This attack is actually a well-known attack, so-called *Alice and Bob Collusion attack (ABC)* [27];

(iii) *An external DoS attacker*, who is able to perform two types of attack, both resulting in Denial of Service (DoS) to the users. The first attack is *expired beacon replay* attack. We consider this attack since in our proposed approach the AP can change the PSK periodically (or after each successful connection, or based on a specific event), and broadcast the new encrypted PSK, inside the beacons, to the users (the whole procedure is explained in Section IV). Hence, if an attacker replays the old expired beacons (i.e., the beacons that contain an old PSK), a user who receives and rebuilds such beacons cannot retrieve the new PSK and cannot access the AP. In the second attack, the attacker's goal is to prevent the legitimate users from connecting to the Wi-Fi. This attack applies to the scenarios in which the AP refreshes the PSK after each successful user connection. Taking advantage of this feature, a malicious user whose attributes satisfy the access policy repeatedly decrypts the PSK and connects to the AP. By doing so, the attacker repeatedly triggers a PSK refresh command at the AP, which leads to preventing the other users to successfully receive the PSK and access the AP.

(iv) *An external brute-force attacker*, who does not satisfy the access policy enforced on the ciphertext and performs a brute-force attack to obtain the PSK;

(v) *A revoked user*, whose attribute-based secret key is revoked, but she is trying to decrypt the PSK and access the Wi-Fi. The revocation of the key could be due to the expiration of his attributes (e.g., a graduate student is not supposed to have a secret key for the *"Student"* attribute) or misbehavior of the user (e.g., the user leaked the attribute's key).

IV. Proposed Approach

We now present our multi-authority attribute-based access control approach for WLANs in two parts: (1) the AP side procedure, and (2) the user side procedure. In Figure 2, we provide an example scenario of our proposal. In this figure, we consider three users (borrowed from Figure 1) who intend to access the Wi-Fi AP, each of which is assigned attribute-based secret keys from one or more authorities as explained in Section II.

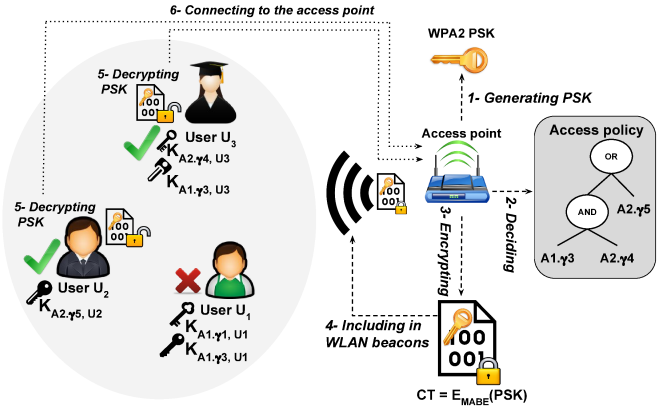

Fig. 2. An example scenario of the proposed approach

(1) *The AP side:* the Wi-Fi AP generates a random WPA2-PSK that protects access to Wi-Fi (Step 1 in Figure 2). Then, it generates an access matrix (A, π) based on which defines the access policy π over the users' attributes (Step 2 in Figure 2). In the next step, the AP runs multi-authority CP-ABE *Encryption* algorithm [9] (explained in Section II-A2) and takes as input the PSK, the access matrix (A, π) over specific attributes, the public keys of the relevant authorities (related to the considered attributes), and the global parameters. The AP encrypts the PSK and outputs $CT = E_{MABE}$(PSK) (Step 3 in Figure 2). Then, the AP embeds the CT in the IEEE 802.11 information elements of the beacons (for information about the beacon management frame please refer to [28]) and sends to the users in its range (Step 4 in Figure 2).

(2) *The user side:* a user U_i who intends to access the WLAN captures beacons (by scanning or sniffing) from the access point AP. Extracts the IEEE 802.11 information elements containing CT. Checks if a subset of the secret keys associated to his attributes satisfies the policy π on the CT. If yes, he runs multi-authority CP-ABE *Decryption* algorithm [9] (explained in Section II-A2), i.e., PSK= $D_{MABE}(CT, GP, \{K_{\gamma_i,U_1}\})$ and obtains PSK (Step 5 in Figure 2). In the example scenario in Figure 2, only U_2 and U_3 are able to satisfy the policy. Subsequently, the user connects to the AP using the obtained PSK (Step 6 in Figure 2). After being successfully authenticated, the user and the AP negotiate on a unique fresh session key K_s which will be used for further communication encryption.

We propose the usage of PSK by the WPA2 protocol as a per-client pre-shared key which can be used only one time by a user, however PSK refreshing could be done also periodically or event-based. As soon as *one* user successfully connects to the AP, the AP performs the above-mentioned procedure (1) for the next user, generating and distributing *a new randomly generated* PSK. It is worth mentioning that, an important feature of our proposal is that, it enables the AP to refresh and randomize the PSK periodically, or per connected user. While, in the traditional Wi-Fi access control mechanisms, the pre-shared secret key refreshing is a challenge, as explained in Section I. Note that, due to the features of the WPA2 protocol, when the AP generates a new PSK, the users which are already connected to that access point will still stay connected. This is due to the fact that PSK is only used for authentication, while further communication between the user and the AP is secured with an agreed session key K_s.

In order to transmit the CT to the users, similar to WI-FAB [7], we can use fountain coding (please refer to [29]) to encode the CT in the IEEE 802.11 information elements. Information elements allow a maximum payload size of 255 bytes, but the CT can easily exceed this limit. To address this issue, we divide the CT into smaller chunks, encode it into *droplets* through fountain coding, and broadcast them. The users who are willing to obtain the CT, need to capture enough of these droplets and use fountain (de)coding to integrate these droplets and reconstruct the original CT. After this step, the user is able to use the multi-authority CP-ABE decryption algorithm (if his attributes satisfy the access policy) and recover the PSK.

A. Security Analysis

In this section, referring to the adversary model we considered in Section III-B, we discuss the security of our proposed scheme against each of the considered adversaries.

a) Passive attacker: considered passive attackers are:

(i) *External eavesdropper:* the proposed scheme is secure against such an attacker, since if the attacker is not able to satisfy the access policy specified on the encrypted PSK, she is not able to decrypt and obtain the PSK. The security of the scheme relies on the security of the multi-authority CP-ABE against polynomial time attackers, for the proof please refer to [9].

(ii) *Honest but curious AP:* the proposed approach does not reveal any sensitive information about the users, i.e., users' credentials and their identities. The only information about a user that could be accessed by the AP is a set of user's attributes that satisfies the access policy. However, this set of attributes highly depends on the combination of attributes defined in the access policy. For example, if the access policy is in the form of $\langle \gamma_1 \ OR \ \gamma_2 \rangle$, then the user who has satisfied the access policy might have γ_1 or γ_2 with the same probability. However, if the access policy is in the form of $\langle \gamma_1 \ AND \ \gamma_2 \rangle$, then the user for sure has both γ_1 and γ_2 attributes. Another point here is that mapping between the attributes and the users' identities, is not trivial. Since the user never reveals his identity, and there is no clear bound between the users' identity and attributes, the AP is neither able to link the attributes to the users' identity, nor identify the users.

(iii) *Honest but curious credential issuing authority:* the proposed approach preserves the privacy of the users against such an attacker, since the AP authorizes the users offline

and the issuing authority is not involved in the user authorization procedure at the AP.

(iv) *Internal eavesdropper:* the proposed approach is secure against such an attacker and does not violate the privacy of other connected users. This is due to the fact that each user who can satisfy the access policy is only able to retrieve the PSK key which is used to connect to the AP. While further communication between the AP and the users is encrypted using a session key, K_s, which is unique per user.

b) Active attacker: In the following, we discuss the security of our proposal against considered active attackers:

(i) *Set of colluding users on the attributes:* relying on the multi-authority CP-ABE scheme [9], our proposed approach is secure against collusion attack. This is due to the usage of the users' global identity (GID), which binds the user's attributes to that specific user. In particular, in multi-authority CP-ABE, the encryptor blinds the message with some shares of a secret, and the decryptor can recover the blinding factor if and only if a set of keys associated to his $\langle GID, attribute \rangle$ pair satisfies the policy. Since two colluding user will have different GID, they can neither recover the blinding factor, nor decrypt the message. For more details please refer to [9].

(ii) *Set of colluding users on the PSK:* this attack does not only relate to our protocol or Wi-Fi access control, but it also applies to any other kind of password-based access control on any data/resource. In particular, in any secured system with a password, there could be a user who successfully passes the authentication checks, obtains the key, and shares the key with other colluding users. Our approach is not safe against such an attacker.

(iii) *External DoS attacker:* in our proposed scheme, similar to all the other Wi-Fi access control schemes, DoS on the users is unavoidable. Considering both attack scenarios (i.e., message replay and key refresh triggering attacks), refreshing the PSK could be done either after every single connection or periodically (even due to a specific event). Actually, in our proposed method, the admin of the AP can perform a trade-off between the connection delay and security by adjusting the suitable time to refresh the PSK. Moreover, going back to the second attacker (i.e., in the key refresh triggering attack), all the users (legitimate or attacker) have more or less the same probability of getting the beacons and decrypting the key.

(iv) *An external brute-force attacker:* in order to protect the system against such an attacker, we propose to generate a random key, of the maximum size allowed by WPA2, and change it either after each user connection, or periodically. This makes it difficult for an attacker to guess the key, assuming the usage of a secure Random Number Generator (RNG) algorithm in the system.

(v) *A revoked user:* Our proposed method does not actually address this issue, since the hardness of denying access of a revoked user relies on the difficulty of revoking a user in an ABE-based system (i.e., revoking the attributes associated to that user as proposed in [30]). Such an attribute revocation approach is neither practical nor scalable, since it requires an authority to periodically broadcast a key update information so that only the non-revoked users can update their keys and continue to decrypt messages.

V. Implementation and Results

Our implementation is targeted, on the access point side, at running on low-end routers equipped with operating systems for embedded devices such as OpenWrt (a Linux-based distribution for routers) or LEDE (an active fork of OpenWrt) [31].

One of the main contributions of this paper is the successful port of the Python Charm framework [32] (together with the GMP and PBC libraries) to the MIPS architecture. We provide openly available package feeds for the firmware build systems of LEDE and OpenWrt[2].

Our implementation of the proposed approach did not require modifications to the OS kernel: the current Linux mac80211 architecture already exposes the needed APIs (nl80211) to the user space. These APIs are employed on Linux-based operating systems by tools such as *wpa_supplicant* (station side) and *hostapd* (access point side). We instead changed slightly (a 1-line patch) the *wpad-mini* daemon (a stripped down version of hostapd [33], included by default in LEDE and OpenWrt) to avoid disconnecting the stations when reloading the configuration file. Indeed, wpad-mini can be instructed, through a HUP signal, to re-read at run-time its configuration file. Moreover the configuration of wpad-mini may include a parameter named `vendor_elements` which allows to specify, through a hexadecimal string, the content of the vendor specific information elements of the IEEE 802.11 beacons [28]. We actively make use of these features of wpad-mini to update the information in the beacons at run-time. To this aim we have developed a Python user-space daemon which runs also on the access point and controls the operation of wpad-mini. This daemon requires the CP-ABE public parameters and an attribute-based access policy for the WLAN in order to perform the following operations: (i) generates a random WPA2 secret, encrypts it with CP-ABE using the provided policy and serializes the ciphertext. This operation is performed both for single-authority CP-ABE and multi-authority CP-ABE using the Charm framework; (ii) divides the ciphertext into droplets (see Section IV); (iii) cycling through the chunks: the chunk is converted into a hexadecimal string. This string is used to update the `vendor_elements` parameter of the wpad-mini configuration file. A HUP signal is sent to the running wpad-mini to instruct it to re-read its configuration; (iv) depending on the settings, the above steps are repeated either when a station associates to the access point or at regular intervals.

[2] https://github.com/netgroup/wifab-openwrt

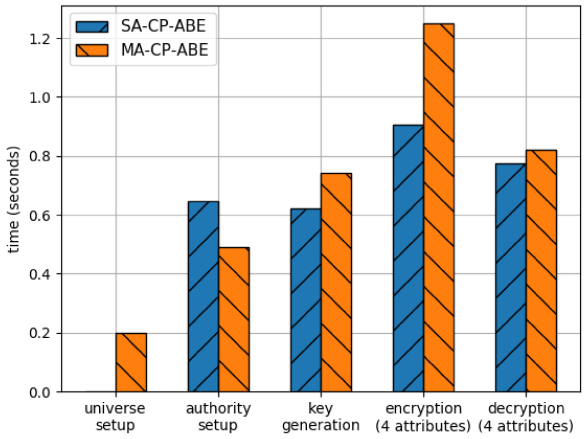

Fig. 3. Average times of main cryptographic operations executed on the router (a low-end MIPS device) using the ported Charm framework

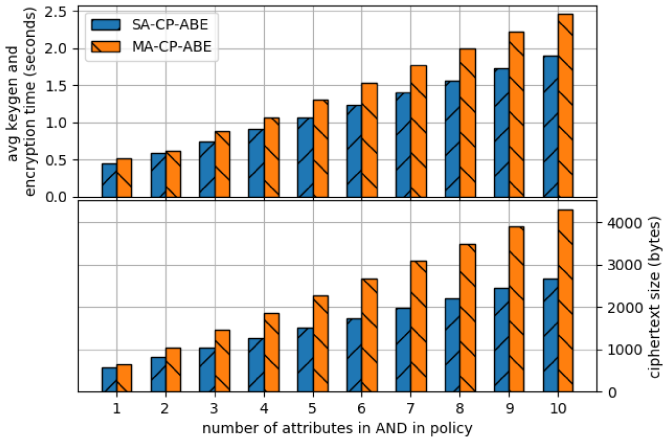

Fig. 4. Average key generation and encryption times on the router vs. number of ANDed attributes in policy and ciphertext size vs. number of ANDed attributes in policy

A. Experimental Results

To test the feasibility of our approach in a real-world scenario, we employ off-the-shelf consumer-grade hardware. Specifically, we use a Netgear R6100 router (Atheros AR9344 SoC, 560 MHz CPU, 128 MB RAM, 128 MB Flash) as access point, while Lenovo T450S laptops are employed as clients/stations. On the router we install LEDE and the software components as described above in Section V.

The performance of the main single-authority CP-ABE [8] (here referred as SA-CP-ABE) and multi-authority CP-ABE [9] (here referred as MA-CP-ABE) operations executed on the router are summarized in Figure 3. The depicted times are averaged over 30 runs. Please note that SA-CP-ABE does not include a universe setup operation. Our results show that a low-end embedded device with a slow CPU can perform most of the considered operations in at most a few seconds.

For our scenario we are especially interested in the performance of the WPA2 key generation and encryption, as these operations are performed on the access point. Figure 4 shows the average key generation and encryption times as well as the size of the generated ciphertext vs. the number of attributes in the policy. A random key is generated and then encrypted using policies with an increasing number i of attributes (from 1 to 10). The policies that we are using for the experiments are in the form $\langle \gamma_1 \; AND \; \gamma_2 \; AND \; \ldots \; AND \; \gamma_i \rangle$, as this form provides the worst case scenario from the computational point of view. The results are averaged over 30 runs. Please note that as the information elements may contain a limited payload (255 bytes), the size of the ciphertext affects the number of chunks/droplets that are required to reconstruct the secret.

Figure 5 illustrates the average time (over 25 runs) needed by the clients/stations to connect to the access point using the proposed approach. In the first phase, the chunks included in the beacons are collected to reconstruct the ciphertext and the ciphertext is decrypted. Then the decrypted WPA2-PSK is

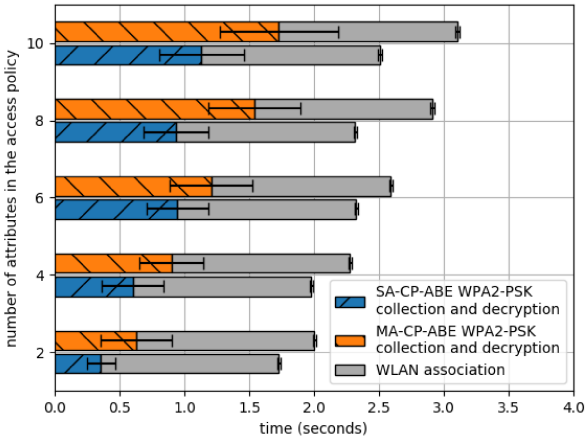

Fig. 5. Required client connection times vs. number of attributes

used to perform the standard WPA2 WLAN association. As it can be seen in the figure, the overhead added to the WLAN association times is comparable with the average time used to connect to WPA2-protected access points and usually accepted by Wi-Fi users. In other words, our proposal does not impose too much delay for connecting to the AP, compared to the traditional method where our proposal is not in place.

VI. Discussion and Conclusion

In this paper we proposed a multi-authority attribute-based access control mechanism for WLANs. In the proposed approach, which is an extension of our previous work WI-FAB [7], adopting *decentralized multi-authority CP-ABE* [9], we facilitate the access control for scenarios in which the users of the WLAN are issued attribute-based credentials from multiple domains/authorities. The advantages of the proposed approach are: (i) authentication of the users neither relies on a

central authority, nor on an online backend infrastructure; (ii) it preserves the privacy of the users by authenticating them based on their profile attributes, not their identity; (iii) it addresses the PSK refreshing issue of the traditional WLAN access control systems. Considering an extensive attack model, we discussed and proved the security of our proposal. Moreover, our real-world experiments demonstrate the feasibility of our approach using off-the-shelf low-end embedded hardware, without requiring major (i.e., kernel-space) firmware modifications. Our results show that the overhead added by the chunk collection and key decryption phase is of the same magnitude of the time required for standard WLAN association.

The main challenges of our proposed scheme, which we leave as future work are: 1) the scalability of the system, and 2) the revocation of the users. The challenge of scalability comes from the fact that the same attribute (e.g., *"student"*) issued by different authorities (e.g., $\mathcal{A}_1$ and $\mathcal{A}_2$) is treated as different attributes (i.e., $\mathcal{A}_1$.student, $\mathcal{A}_2$.student), which is not desirable in large scale scenarios. Actually, we anticipate that this issue could be mitigated by supporting the aggregation of such attributes. Considering the user revocation challenge, different from other use cases of ABE (e.g., access control on the encrypted data, in which the data owner does not have control over the encrypted data after publishing it), in WLAN scenario a backend infrastructure is in place. Therefore, a possible promising solution for user revocation could be two-phase authorization system: in the first phase, adopting our proposed approach, the user who satisfies the access policy connects to the Wi-Fi infrastructure. In the second phase, the user needs to prove that his privileges have not been revoked, for which we could adopt Certificate Revocation List (CRL) in the system, along with anonymous credential systems such as *Idemix* [25] or *U-Prove* [26].

ACKNOWLEDGMENT

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

REFERENCES

[1] (2016) ict facts and figures 2016. http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx.

[2] (2017) Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[3] A. Cassola, E.-O. Blass, and G. Noubir, "Authenticating privately over public Wi-Fi hotspots," in *Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS'15. ACM, 2015, pp. 1346–1357.

[4] WiFi Alliance, "WPA2 security now mandatory for Wi-Fi certified products," *Press Release*, 2006.

[5] H. Boland and H. Mousavi, "Security issues of the IEEE 802.11b wireless LAN," in *Proc. of the Canadian Conference on Electrical and Computer Engineering*, vol. 1. IEEE, 2004, pp. 333–336.

[6] A. Cassola, W. K. Robertson, E. Kirda, and G. Noubir, "A practical, targeted, and stealthy attack against WPA enterprise authentication." in *Proc. of the 20th Annual Network and Distributed System Security Symposium*, ser. NDSS'13, 2013.

[7] C. Pisa, A. Caponi, T. Dargahi, G. Bianchi, and N. Blefari-Melazzi, "WI-FAB: attribute-based wlan access control, without pre-shared keys and backend infrastructures," in *Proc. of the 8th ACM International Workshop on Hot Topics in Planet-scale mObile computing and online Social neTworking*. ACM, 2016, pp. 31–36.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. of the IEEE Symposium on Security and Privacy*, ser. SP'07. IEEE, 2007, pp. 321–334.

[9] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–EUROCRYPT*. Springer, 2011, pp. 568–588.

[10] K. Frikken, M. Atallah, and J. Li, "Attribute-based access control with hidden policies and hidden credentials," *IEEE Transactions on Computers*, vol. 55, no. 10, pp. 1259–1270, 2006.

[11] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control," *Computer*, no. 2, pp. 85–88, 2015.

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

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

[14] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. of the 14th ACM conference on Computer and communications security*. ACM, 2007.

[15] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[16] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M. Rahmani, and P. Liljeberg, "On the feasibility of attribute-based encryption on internet of things devices," *IEEE Micro*, 2016.

[17] J. Ning, Z. Cao, X. Dong, and L. Wei, "White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[18] M. Jo, V. Odelu, A. K. Das, M. K. Khan, and K.-K. R. Choo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, 2017.

[19] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*. Springer, 2007, pp. 515–534.

[20] S. Muller, S. Katzenbeisser, and C. Eckert, "On multi-authority ciphertext-policy attribute-based encryption," *Bulletin of the Korean Mathematical Society*, vol. 46, no. 4, pp. 803–819, 2009.

[21] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proc. of the International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 315–332.

[22] P. Arana, "Benefits and vulnerabilities of Wi-Fi protected access 2 (WPA2)," *INFS 612*, pp. 1–6, 2006.

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

[24] J. Camenisch, M. Dubovitskaya, K. Haralambiev, and M. Kohlweiss, "Composable and modular anonymous credentials: definitions and practical constructions," in *Proc. of the International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2014, pp. 262–288.

[25] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proc. of the 9th ACM conference on Computer and communications security*. ACM, 2002.

[26] C. Paquin and G. Zaverucha, "U-prove cryptographic specification v1. 1," *Technical Report, Microsoft Corporation*, 2011.

[27] (2016) OAuth: the ABC attack (the alice and bob collusion attack). https://www.ietf.org/mail-archive/web/oauth/current/msg16767.html.

[28] "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*, pp. 1–1076, June 2007.

[29] 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*, vol. 28, no. 4, pp. 56–67, 1998.

[30] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 199–217.

[31] "LEDE project," https://lede-project.org/.

[32] "Charm: A framework for rapidly prototyping cryptosystems," https://github.com/JHUISI/charm.

[33] "Linux wireless - hostapd," http://linuxwireless.org/en/users/Documentation/hostapd/.