# A Smart-Phone Security Framework for Accessing Enterprise Wi-Fi Networks

Sorin ŢEICAN, Bogdan-Cosmin CHIFOR, Mihai TOGAN, and George GUGULEA

*Abstract*—**The authentication and authorization process is a critical factor in Wi-Fi enterprise networks, being the mechanism used to give access to the network resources. The general Wi-Fi enterprise network security frameworks have a static structure, being based on user-name/password credentials and without offering a flexible granular access. Nowadays, smart-phones devices are becoming ubiquitous and these can be used as security elements due to their hardware/software capabilities. Smart-phones have the capability to execute complex cryptographic operations and have the hardware peripherals, which allow authenticating the user. In this paper, we propose a smart-phone centric authentication and authorization framework which employs various bootstrap security protocols to provision temporary credentials for a WPA2 Enterprise protected Wi-Fi network.**

*Index Terms*—**smart-phone, security, FIDO, RADIUS, Wi-Fi, Android.**

## I. Introduction

The security mechanisms are a critical factor for a Wi-Fi enterprise network because the resource owner must give access only to authenticated and authorized users. Along with the increased popularity of smart-phone devices, there are new requirements for this type of networks: flexibility, granular access, revocation and ease of use. Thus for environments like university campuses or companies which offer a Wi-Fi resource to visitors or temporary collaborators, a flexible security network access is mandatory, in order to address a custom application use-case.

Currently the Wi-Fi security solutions have a static structure, most of them employing a WPA2 Enterprise mechanism with the 802.1X protocol authenticating and authorizing the users which are provided user name/password credentials. WPA2 Enterprise solutions use a back-end RADIUS server to store the user credentials and other attributes, the authentication response being relayed to the Network Access Server (NAS), which allows the supplicant to access the network resources.

This work has been partially funded by the European Commission in part of the ReCRED project (Horizon H2020 Framework Programme of the European Union under GA number 653417).

S. Ţeican is with the certSIGN, Research and Development Dept., 29 A Tudor Vladimirescu Ave., AFI Tech Park 1, District 5, Bucharest, Romania, 050881

B.-C. Chifor is with the certSIGN, Research and Development Dept., 29 A Tudor Vladimirescu Ave., AFI Tech Park 1, District 5, Bucharest, Romania, 050881

M. Togan is with the Military Technical Academy, Computer Science Dept., 39-49 George Coșbuc Ave., Sector 5, 050141, Bucharest, Romania (e-mail: mihai.togan@mta.ro).

G. Gugulea is with the certSIGN, Research and Development Dept., 29 A Tudor Vladimirescu Ave., AFI Tech Park 1, District 5, Bucharest, Romania, 050881

This type of architecture is suitable for an environment where the supplicants have a well-defined set of access attributes and follow a strict security policy. When authorizing temporary users, the NAS must be integrated with a remote security domain.

This security domain handles the user attributes and the authorization process is executed in a federated manner. Also in this type of scenario, the user's smart-phone is used as supplicant device, thus the security system must provide both user to device authentication and device to service authentication, while provisioning the smart-phone with temporary access credentials, which must be delivered to the NAS.

In this paper, we propose a smart-phone centric security solution, which completes the WPA2 Enterprise scheme by executing a bootstrap authentication protocol, which allows the smart-phone application to be provisioned with temporary Wi-Fi credentials.

We address the scenario of a university campus or a company client network, where the visitors use the smart-phone as security element to access a Wi-Fi protected resource: an e-learning or a confidential business website. For this scheme, we use FIDO as authentication protocol, which runs on the smart-phone, and we propose a security stack, which comprises several server-side modules. The rest of the paper is organized as follows. In Section 2 we present the related work. Sections 3 and 4 present the OpenID Connect (OIDC) and Attribute based Access Control (ABAC) system architecture, describing each module of the proposed security stack. In Section 5 we present details regarding the system implementation: the NAS and the Android security application.

Finally, Section 6 outlines our conclusions. This is an extended paper of the article entitled "A flexible authorization mechanism for enterprise networks using smart-phone devices", published at the COMM 2018 conference.

## II. Related Work

The smart-phone can be used as security element because it has connectivity capabilities and it is equipped with hardware modules that permit the user identification (e.g. fingerprint reader, facial recognition using the camera). The idea of using a smart-phone as authentication element is currently explored by both academia and commercial products.

In [1], Shafique et al. describe a series of attacks and vulnerabilities of smart-phone authentication mechanisms and present a comparative analysis of various authentication

Digital Object Identifier 10.32754/JMT.2018.2.03

techniques.

Agrawal and Patidar explore in [2] multiple smart-phone security techniques addressing the user-to-device authentication using biometrics behavioral analysis. Regarding the importance of eliminating the passwords in the authentication process, Lindemann presents in [3] the advantages of FIDO, a challenge-response protocol which employs cryptographic keys.

In [4], Everts et al. also explore the idea of using a smart-phone to authenticate to web-services by means of cryptographic keys, thus replacing the passwords. The security features of Android enabled smart-phones and their capability of executing cryptographic operations in a Trusted Execution Environment (TEE) are analyzed by Cooijmans et al. in [5].

In [6], Broeder et al. present the advantages, in terms of security and operational costs, when a federated identity management system is employed by multiple organizations. The importance of using authentication mechanisms (for accessing both web and physical resources) which take into consideration the usability factor is stressed by Carullo et al. in [7]. Multiple aspects regarding the Guest Wi-Fi security, are described by Yap et al. in [8], presenting the problem of non-uniform authentication methods for this type of networks.

## III. OIDC System Architecture

The core element of our system architecture is the authentication and authorization stack comprised of the following: FIDO UAF & OpenID Connect as services and as client code libraries residing on the user device application. The proposed security system architecture is based on the work done in [10-12].

FIDO UAF [9] is employed for the user to device and device to service authentication part of the security system, and OpenID Connect as authentication federation and authorization mechanism for organization services.
FIDO UAF on the user device takes advantage of new Android features such as the Keystore that allows storing cryptographic material (asymmetric private keys in our case) on the Trusted Execution Environment of the device that is authenticated using biometric identification mechanisms such as fingerprint, or face recognition.

As for the federated authentication part, FIDO UAF and OpenID Connect can easily be extended to communicate with each other. This extension allows the user to register his biometric attributes instead of cumbersome and complex passwords that in many cases are reused, or forgotten.

User biometrics are used to access his cryptographic credentials and never leave his device through the whole authentication and authorization process.

Generating RADIUS credentials for the Auth Wi-Fi SSID can easily be achieved by extending the FIDO UAF Server to generate and save such credentials inside the RADIUS database after a successful authentication process for a particular user.

As observed from the proposed system architecture diagram (Fig. 1), there are four steps required in order to authenticate and authorize service access to the user:

1. User enrollment;
2. The user authenticates with FIDO UAF which generates the RADIUS credentials that are transferred through TLS to the user device and triggers the Android Wi-Fi Manager to switch and authenticate to the Auth Wi-Fi SSID;

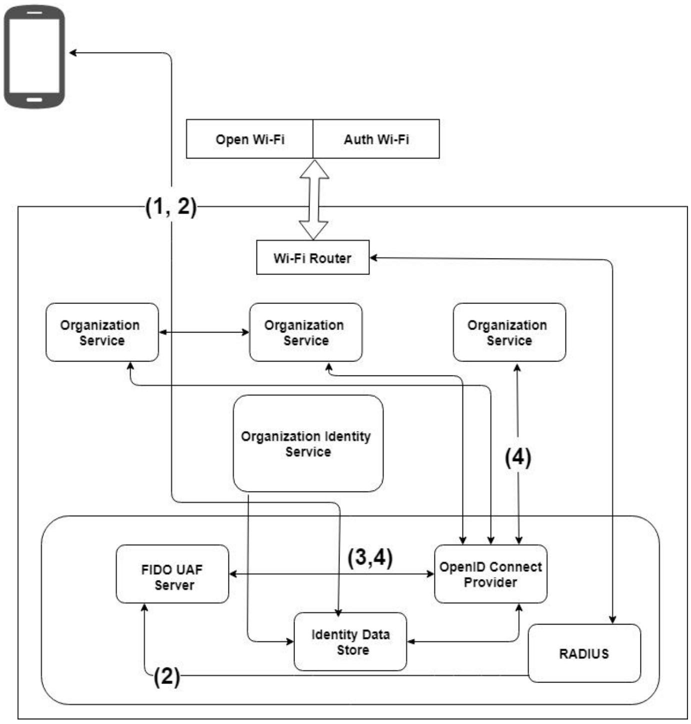


Figure 1. General system architecture

3. Besides generating RADIUS credentials, the FIDO UAF Server also generates an authentication ID, which is transferred to the user device and is sent to the OIDC Provider in order to verify authentication status.
   This authentication ID is designed as a short lived token which is consumed (invalidated) at every successful validation response by the FIDO UAF Server, or after a designated time frame;
4. After the user selects an organization service, the OIDC authorization process starts and the Identity Provider verifies with the FIDO UAF Server if the user is authenticated and if his attributes satisfy the authorization policy.

After an operator registers the visitor, the Organization Identity Service provisions OpenID Connect Provider Identity Data Store and sends the credentials through e-mail to the visitor.

To prevent malicious registrations, the FIDO UAF Server is tightly coupled with the OpenID Connect Provider acting as Service Provider. Instead of sending the username as in the normal FIDO UAF registration sequence in order to receive the registration request from the server, the FIDO UAF Client sends the access token received from authenticating with the OIDC IP (here considered as an activation process using the credentials received through the enrollment e-mail). After receiving the access token through HTTPs from the FIDO UAF Client, the server acquires the username from the OIDC IP and generates the registration request for the client, thus blocking unauthorized users.

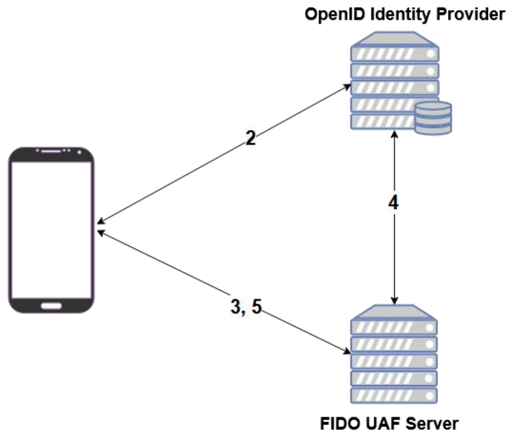


Figure 2. Wi-Fi authentication system enrollment

The Wi-Fi authentication system enrollment presented in Fig. 2 has the following phases:

1. user receives enrollment email;
2. user authenticates and receives access token from OIDC IP;
3. FIDO UAF Client sends access token to the FIDO UAF Server in order to receive the registration request;
4. the FIDO UAF Server uses the access token in order to acquire username;
5. normal FIDO UAF registration protocol is initiated.

The authorization stack also allows internal users such as employees to authorize internet access to visitors. The non-authenticated SSID (Open-Wifi) is configured with a custom captive portal that verifies connected devices MAC addresses with a QR Authenticator and routes the device either to *Internet* or to the internal security bootstrapping service. The unauthorized visitor connects to the Open-Wifi and is redirected to the captive portal which then presents a QR code which is linked to his MAC address.

An authorized employee scans this QR code with his device, which contains an ID and the QR Authenticator address. The authorized device sends this ID together with the OIDC access token to the QR Authenticator which verifies the identity of the authorized user and then allows the *Internet* access, authoring the user device MAC address.

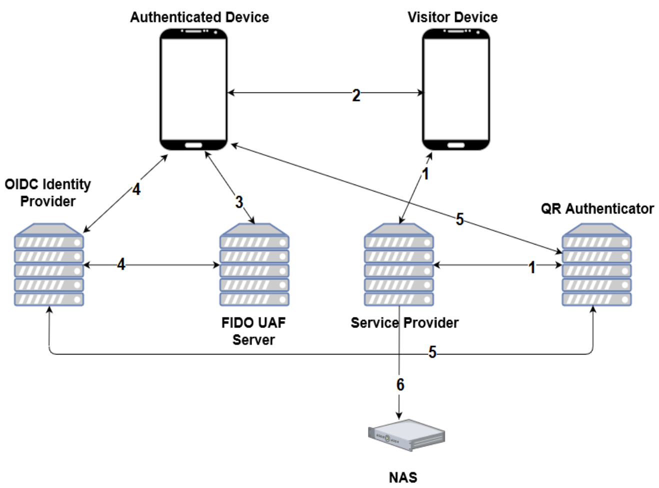


Figure 3. QR based Wi-Fi authentication

1. The visitor device obtains the QR code from the QR Authenticator through the Service Provider which initiates the polling mechanism and records the visitor MAC address;
2. The authenticated device scans the QR code which contains the request ID and QR Authenticator location;
3. The device authenticates with FIDO UAF obtaining the FIDO authentication ID;
4. The device requests an OIDC access token with the FIDO authentication ID;
5. The authorized device sends the QR request ID together with the access token to the QR Authenticator that verifies the token and authorizes the QR request ID;
6. Through the polling mechanism, the Service Provider acknowledges the authorization of the request and adds the visitor MAC address to the NAS firewall.

The client device MAC address is obtained by querying the NAS local ARP cache table, using the client device IP address delivered by the HTTP server (obtained from the HTTP socket).

## IV. ABAC System Architecture

In order to integrate the system in public institutions where user tracking is prohibited, the system architecture can be modified to use attribute based access control (ABAC) in order to preserve anonymity. The OIDC authorization mechanism can be replaced by U-Prove or Idemix protocols. To authenticate the user to the U-Prove/Idemix token issuance service, the organization identity service provisions the issuance service with user attributes and sends an activation code to the user, which is used during the issuance protocol in order to receive verified attributes tokens. A simple Service Provider replaces FIDO UAF in order to generate RADIUS credentials based on token attributes checked with the U-Prove/Idemix verification service.

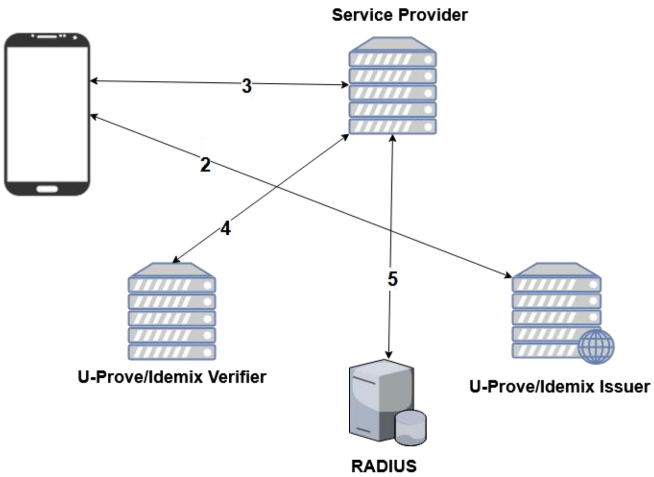


Figure 4. P-ABAC based Wi-Fi authentication

The P-ABAC Wi-Fi authentication scheme presented in Fig. 4, consists in the following steps:

1. user enrollment;
2. user receives attributes token based on the activation code;
3. user requests *Internet* access using U-Prove/Idemix token;
4. service Provider verifies U-Prove/Idemix token with the Verifier service;
5. if token passed the verification, the Service Provider generates RADIUS credentials and sends them to the user device through HTTPs.

The functionality that allows authenticated users to authorize *Internet* access for visitors consists of configuring the QR Authenticator to check the attribute token of the authorizing user with the U-Prove/Idemix verification service.

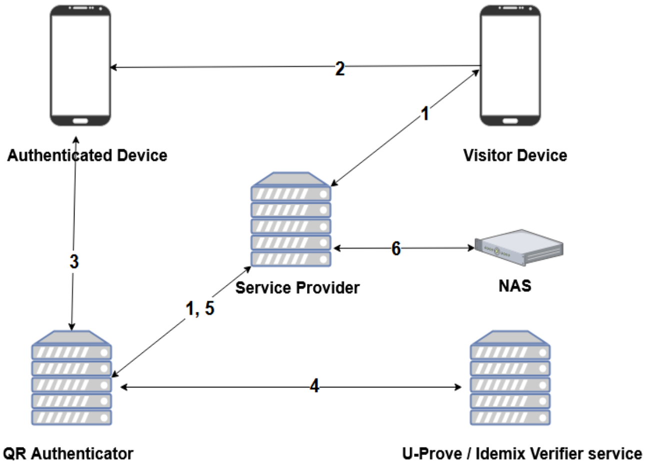


Figure 5. P-ABAC QR Wi-Fi authentication

The P-ABAC QR based authentication presented in Fig. 5 has the next steps:

1. Visitor requests QR code from the Service Provider which in turn receives it from the QR Authenticator;
2. The authenticated device scans the QR code;
3. The authenticated device redirects it together with its attribute token to the QR Authenticator;
4. The QR Authenticator verifies the identity of the authenticated user;
5. During steps 2, 3, 4 the Service Provider polls the QR Authenticator to check the authorization status of the scanned QR code.

## V. System Implementation

Regarding the system implementation, for the Android security application we used a smart-phone equipped with a fingerprint reader, which authenticates the user in the FIDO protocol context. For the Wi-Fi router we used a dual-band Linksys WRT 1900 ACS device on which we installed an OpenWRT operating system, this being the NAS for our system. The NAS advertises two Wi-Fi SSID: *WiFi-Open* – the no password network which allows the user to authenticate and obtain the temporary credentials for the *WiFi-Auth* – the WPA2 Enterprise network which exposes the protected resource (in our case this being the Internet connection). Along with the Wi-Fi networks, the employed NAS also has an Ethernet switching module, used to connect the Wi-Fi networks to the server, which runs our authentication stack, and to the Internet gateway. The NAS has four networks, logically separated by VLANs, as follows:

- *WiFi-Open (VLAN O)* – hosts the unauthenticated users, and the server, which offers the authentication services
- *WiFi-Auth (VLAN A)* – hosts the authenticated users, which can access the protected resources
- *Management (VLAN M)* – it is special network which allows managing the NAS via a SSH connection
- *Internet (VLAN I)* – it is the network which offers access to the protected resource. The NAS has a route, which allows the packets originating from the *WiFi-Auth* network to be routed to the *Internet* network (NAT).

The system configuration on the NAS side is presented in Fig. 6. As it can be observed, the NAS Ethernet switching module shares the same VLAN broadcast domain with the Wi-Fi networks, by using Linux bridges (brctl). The NAS CPU port, which is connected to the Ethernet switching module, operates only with VLAN tagged packets in order to forward the packets to the corresponding Linux VLAN interface. For our implementation, we used only one NAS Ethernet port configured as trunk, the actual connection to the authentication server, management module and Internet gateway being handled by another switch.

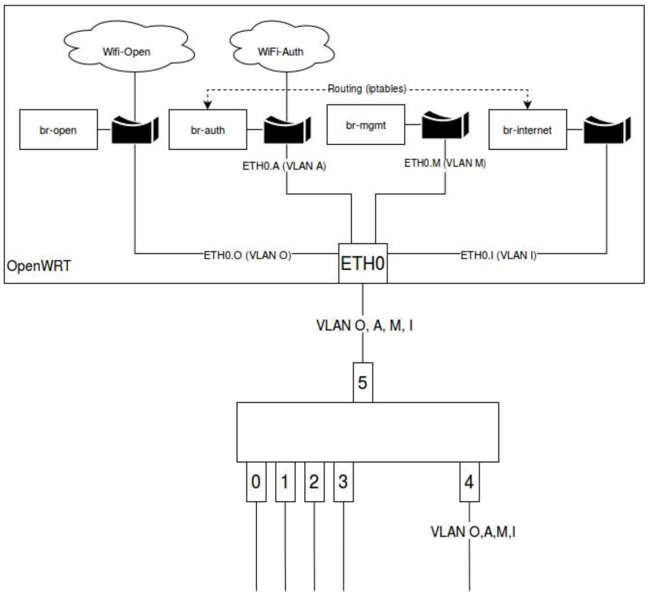


Figure 6. NAS module configuration

The traffic classification is executed on the NAS side, by employing an *iptables* packet marking technique. Thus, when a user is authorized the associated MAC address is installed in a custom iptables PREROUTING chain. If the MAC address of the ingress packet is installed in a PREROUTING iptables rule, an authorization metadata (iptables mark) is attached to the packet. On the NAT table side from the PREROUTING chain, the destination IP address of unauthorized packets is changed to be the IP address of the QR authentication server. Thus, an authorized packet is redirected to the QR authentication server using a captive portal technique. Each iptables rule which classifies the packet as being authorized and bypasses the captive portal, has a limited lifetime. This is achieved using the *comment* feature of iptables rules, by attaching a *comment* with the rule install timestamp. The Service Provider (the entity which enforces the rule on the NAS side), runs a thread which periodically purges the expired filtering rules, thus limiting the user access to sensitive resources and mitigating attacks like MAC cloning of authorized devices.

The packet filtering logic is depicted in Fig. 7.

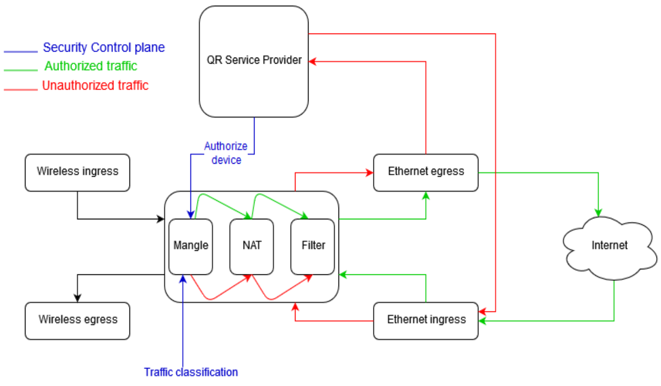


Figure 7. Packet filtering logic

For the advertised Wi-Fi networks, the NAS plays the role of DHCP server, default gateway and DNS server (using OpenWRT dnsmasq), the latter being used to solve to internal domain names employed in the FIDO protocol. The *WiFi-Auth* network has configured as authentication method WPA2 Enterprise, along with a RADIUS back-end server and a shared secret (used by the RADIUS protocol to permit the authentication between the NAS and the RADIUS authentication server). For the RADIUS authentication server we used the FreeRADIUS solution for which we configured an SQL back-end credentials server (MariaDB hosted on the same machine as the RADIUS server) which holds the temporary user credentials. Thus, after the user is authentication on the FIDO server side, the randomly generated user credentials are injected into the SQL back-end. More specifically, we used the FreeRADIUS *radcheck* SQL table to insert the randomly generated credentials as follows: the *UserName* and *Value* fields contain the random material generated by the FIDO server. Taking into consideration this mechanism, the RADIUS server is not aware of the FIDO based authentication stack, executing the authentication session by querying normal user-name/password credentials. This authentication flow is depicted in Fig. 3. The random credentials are short-lived tokens, being erased from the RADIUS database after a few seconds the authentication process is finalized. Thus, the FIDO server controls the time frame in which the authenticated user can use the obtained credentials, mitigating a credential re-use attack at a later time, when the user might be revoked from the system. This mechanism also blocks the attackers that may capture the authentication token from accessing the protected network.

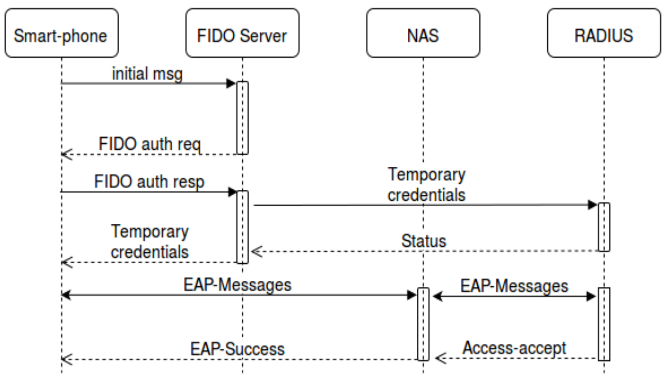


Figure 8. Wi-Fi authentication flow using FIDO on the Android device

On the smart-phone side, after the FIDO authentication ends, the application receives the random credentials that have to be used for the authentication to the protected network via WPA2 Enterprise. In order to design a user-friendly security system, the smart-phone application makes the switch to protected Wi-Fi network automatically, using the provisioned temporary credentials as user-name and password. The Wi-Fi authentication flow on the Android device is presented in Fig. 8.

To implement this feature we used the Android SDK *WifiManager* and *WifiConfiguration* classes to add a new network configuration. In the new Android network configuration, we added the following: the protected network SSID, the key management method is set to *IEEE8021X* and *WPA_EAP*, the EAP method is set to *PEAP* and the random credentials are set as identity and password. Because the 802.1X PEAP method is used, an issue, which must be taken into consideration the RADIUS certificate, trust. This certificate must be verified against a trusted chain or by using a certificate pinning method in order to prevent the smart-phone supplicant to execute the authentication protocol with a rogue RADIUS server, which may capture the random credentials. In our implementation, we installed the RADIUS certificate trusted chain using the previously mentioned Android classes, but in other implementations this certificate could be simply trusted by the Android system or provided by the FIDO server along with the random credentials.

## VI. Conclusions

Flexible authentication and authorization mechanisms are critical for the enterprise networks, which offer services to guest users which may have a temporary access. The security scheme proposed in this paper leverages the security capabilities of Android smart-phones in order to design a user-friendly authentication mechanism the need to provision temporary users with user-name/password credentials or even digital certificates. One of the main advantages of our work is the fact that this system can serve as a basis for a roaming user authentication scenario. Thus, two organizations, which establish a trust relation (mutual authentication), can inter-connect a local Wi-Fi enterprise structure with the FIDO based security stack deployed remotely, obtaining a federated authentication scheme. Another characteristic that must be taken into consideration in this security scenario, is the user anonymity, which may be achieved if the Organization Identity Service and the Wi-Fi enterprise structure are located in separated trust domains. Regarding the user security, the FIDO protocol permits generating a pair of asymmetric cryptographic keys especially for a certain resource (in our case the protected Wi-Fi network), thus the user does not employ other cryptographic material associated with another resource. In addition, the FIDO cryptographic keys are removed and deregistered from the server side after the Wi-Fi usage is ceased. Taking into consideration the modular structure of our security scheme, which may enable federated authentication schemes, rapid used revocation can be obtained even though the user tries to access the resources located at another organization premises. Moreover, the proposed security scheme is feasible in terms of operational expenses because it can be integrated with an already existing Wi-Fi enterprise security structure. Regarding the future work, we plan to adapt the proposed security system to a smart-home scenario, where the user can authorize the IoT device to connect to the Home Area Network (Wi-Fi 802.1X protected). For this scenario, we also plan to address the non-802.1X capable IoT devices by using a MAC authentication bypass mechanism. Another research direction which can be followed is designing an out-of-band authentication mechanism using the proposed FIDO system as a RADIUS extension. Thus, a private 802.1X network which uses RADIUS as authentication mechanism could use the smart-phone based FIDO security mechanism to achieve a multi-factor authentication (e.g. credentials along with smart-phone authentication). The protocol packets between the FIDO server and the client device can be transported by an Internet network, thus allowing the user to access the protected network using another device.

## ACKNOWLEDGMENT

This work has been partially funded by the European Commission in part of the ReCRED project (Horizon H2020 Framework Programme under GA number 653417).

## REFERENCES

[1] U. Shafique, A. Sher, R. Ullah, H. Khan, A. Zeb, R. Ullah, S. Waqar, U. Shafi, F. Bashir, M. A. Shah, "Modern Authentication Techniques in Smart Phones: Security and Usability Perspective," (IJACSA) *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, pp. 331-340, 2017. doi:10.14569/IJACSA.2017.080142

[2] A. Agrawal and A. Patidar, "Smart Authentication for Smart Phones," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 4839-4843, 2014.

[3] R. Lindemann, "The Evolution of Authentication," *ISSE 2013 Securing Electronic Business Processes*, pp. 11-19, 2013, doi:10.1007/978-3-658-03371-2_2

[4] M. H. Everts, J.-H. Hoepman, and J. Siljee, "UbiKiMa: Ubiquitous authentication using a smartphone, migrating from passwords to strong cryptography," in *Proc. ACM Workshop on Digital identity management ACM*, pp. 19-24, 2013, doi:10.1145/2517881.2517885

[5] T. Cooijmans, et al., "Secure Key Storage and Secure Computation in Android," Master's thesis, Radboud University Nijmegen, 2014, doi:10.1145/2666620.2666627

[6] D. Broeder, B. Jones, D. Kelsey, P. Kershaw, S. Luders, A. Lyall, T. Nyronen, R. Wartel, H. J. Weyer, "Federated Identity Management for Research Collaborations," No. CERN-OPEN-2012-006, 2012, doi:10.5281/zenodo.1296031

[7] G. Carullo, F. Ferrucci, and F. Sarro, "Towards Improving Usability of Authentication Systems Using Smartphones for Logical and Physical Resource Access in a Single Sign-On Environment," in *Information Systems: Crossroads for Organization, Management, Accounting and Engineering*, Physica, Heidelberg, 2012. pp. 145-153, doi:10.1.1.717.6561

[8] K.-K. Yap, Y. Yiakoumis, M. Kobayashi, S. Katti, G. Parulkar, and N. McKeown, "Separating Authentication, Access and Accounting: A Case Study with OpenWiFi," OPENFLOW-TR-2011-1, Open Networking Foundation, 2011.

[9] R. Lindemann, D. Baghdasaryan, E. Tiffany (Eds.), "FIDO UAF Protocol Specification v1. 0," FIDO Alliance Proposed Standard, Dec. 2014, https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html.

[10] C. Soriente, et al., "Multifactor authentication for DCA: user-to-device and device-to-network support," ReCRED Project, Deliverable 3.2, 2016.

[11] I. Florea, et al., "Campus-wide Wi-Fi and web services access control pilot set up," ReCRED Project, Deliverable 7.2, 2016.

[12] G. Bianchi, et al., "Full Design and Prototype of the ABAC Infrastructure," ReCRED Project, Deliverable 5.2, 2017.