September 2022

AUTHENTICATION AND KEY MANAGEMENT OF IOT APPLICATIONS WITH EXTENDED WIFI AUTHENTICATION (WIFI AKMA)

Rajesh I V

Ram Mohan R

AUTHENTICATION AND KEY MANAGEMENT OF IOT APPLICATIONS WITH EXTENDED WIFI AUTHENTICATION (WIFI AKMA)

AUTHORS:
Rajesh I V
Ram Mohan R

ABSTRACT

Internet of things (IoT) devices frequently apply insufficient authentication mechanisms with their application servers due to the constrained nature of such devices. For example, most IoT devices lack the resources that are necessary to store usernames and passwords, certificates, and keys in a secured manner. The challenge that was described above is solved in a 3rd Generation Partnership Project (3GPP) fifth-generation (5G) wireless environment through the Authentication and Key Management for Applications (AKMA) initiative. However, there is no AKMA-equivalent facility within a WiFi environment. Accordingly, techniques are presented herein that extend the WiFi authentication process to support application server authentication for constrained devices. Aspects of the presented techniques support an exchange of a WiFi key and a key identifier (which may be referred to herein as a KAKMA key and an A-KID) as part of an Extensible Authentication Protocol (EAP) tunnel using a new information element (IE) once an authentication process has successfully completed. Such an exchange allows a station (STA) device to use the key tuple {KAKMA, A-KID} to access any application functions that are grouped with that key identifier (i.e., A-KID) without requiring any further authentication.

DETAILED DESCRIPTION

Internet of things (IoT) devices frequently apply insufficient authentication mechanisms with their application servers due to the constrained nature of such devices. For example, most IoT devices lack the resources that are necessary to store usernames and passwords, certificates, and keys in a secured manner. Additionally, most IoT device manufacturers ignore the basic security fulfillment of the devices which cause severe security threats during adoption of the devices.

The challenge that was described above is solved in a 3rd Generation Partnership Project (3GPP) fifth-generation (5G) wireless environment through the Authentication and Key Management for Applications (AKMA) initiative. That initiative, which is described initially in 3GPP technical specification (TS) 33.535 release 16 and further enhanced in release 17, focuses on leveraging an operator's authentication infrastructure to secure the communication that takes place between a user equipment (UE) and an application function. Under such an approach, the application server key derivation is based on primary subscriber identity module (SIM) authentication.

However, there is no AKMA-equivalent facility within a WiFi environment. A typical enterprise environment (such as a factory, a healthcare setting, a retail establishment, a mining operation, etc.) will have many IoT devices that make use of WiFi access. Among other things, those devices will need to periodically communicate with their respective application server.

To address the lack that was described above, techniques are presented herein that yield a mechanism that extends the WiFi authentication process to support application server authentication for constrained devices.

Before proceeding with a detailed description of the presented techniques, it will be helpful to confirm two introductory points.

First, wireless clients employ methods that are defined in the Institute of Electrical and Electronics Engineers (IEEE) 802.11x technical standards to authenticate with an access point (AP). Currently, there are many flavors of such authentication. Some of those include the Extensible Authentication Protocol (EAP), the EAP Tunneled Transport Layer Security (EAP-TTLS) protocol, the EAP Flexible Authentication via Secure Tunneling (EAP-FAST) protocol, the EAP Transport Layer Security (EAP-TLS) protocol, and the EAP Protected Extensible Authentication Protocol (PEAP) protocol.

Second, the techniques presented herein employ a number of terms (several of which are borrowed from the AKMA initiative), including:

Term	Description
AAnF	AKMA anchor function
A-KID	AKMA key identifier of a UE
KAF	AKMA application key

KAUSF	AKMA authentication server function (AUSF) key
KAKMA	AKMA Anchor Key - AF_ID application function identifier. These identifiers are used during KAF derivation by both an AF and a UE and are known upfront to a UE and an AF as part of the registration process.

It is important to note that the techniques presented herein may be employed with any EAP variant (several of which were noted above). As will be described and illustrated below, aspects of the presented techniques support the exchange of a WiFi key and a key identifier (which may be referred to herein as a KAKMA key and an A-KID) as part of an EAP tunnel, through the use of a new information element (IE), once an authentication process has successfully completed. Such an exchange allows a station (STA) device to use the key tuple {KAKMA, A-KID} to derive the access keys in order to access any application functions or servers that are grouped with that key identifier (i.e., A-KID) without requiring any other application server specific authentication keys and infrastructure.

While the techniques presented herein are particularly suited to constrained IoT devices, it is important to note that the techniques may be applicable and extendable, as needed, to any WiFi clients.

Figure 1, below, presents elements of an overall architecture that is possible according to aspects of the techniques presented herein and which is reflective of the above discussion.

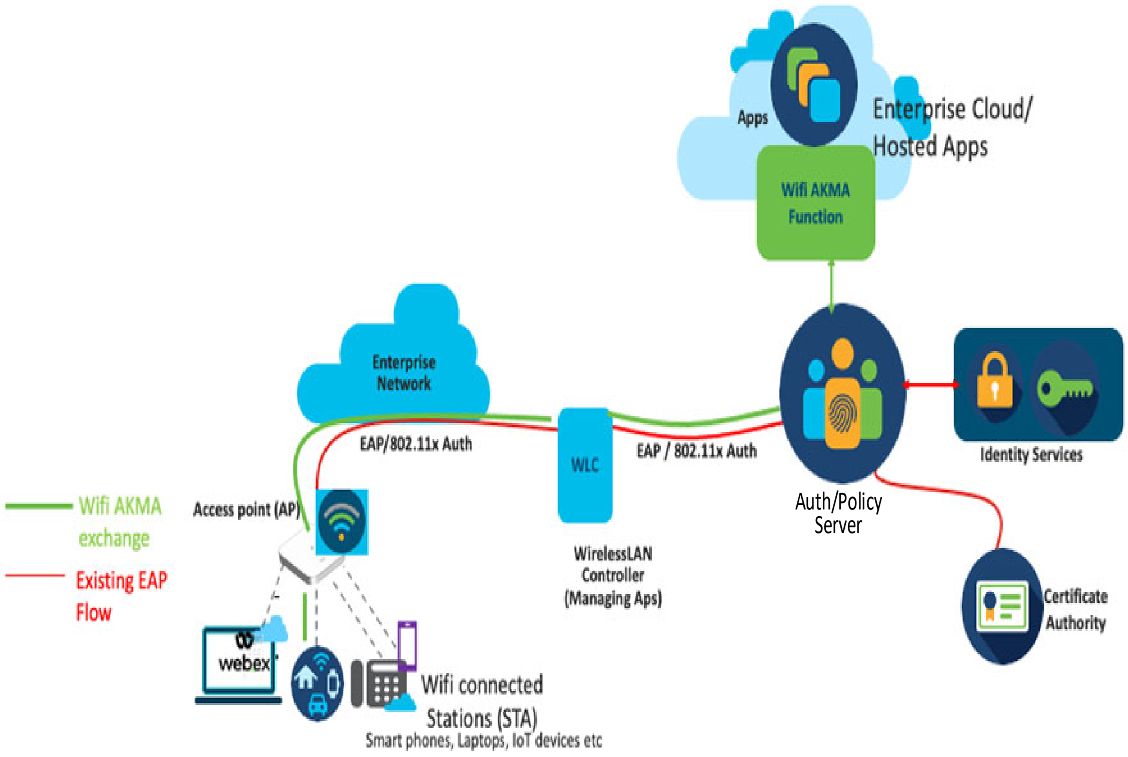

Figure 1: Exemplary Architecture

As depicted in Figure 1, above, the techniques presented herein introduce a new function (which is labeled WiFi AKMA Function in the figure) which is a logical entity that may be part of an authentication/policy (auth/policy)server or that may operate as a separate function.

In support of the upcoming detailed discussion of the techniques presented herein, Figure 2, below, presents elements of an illustrative sequence diagram that is possible according to aspects of the techniques presented herein and which is reflective of the above discussion.

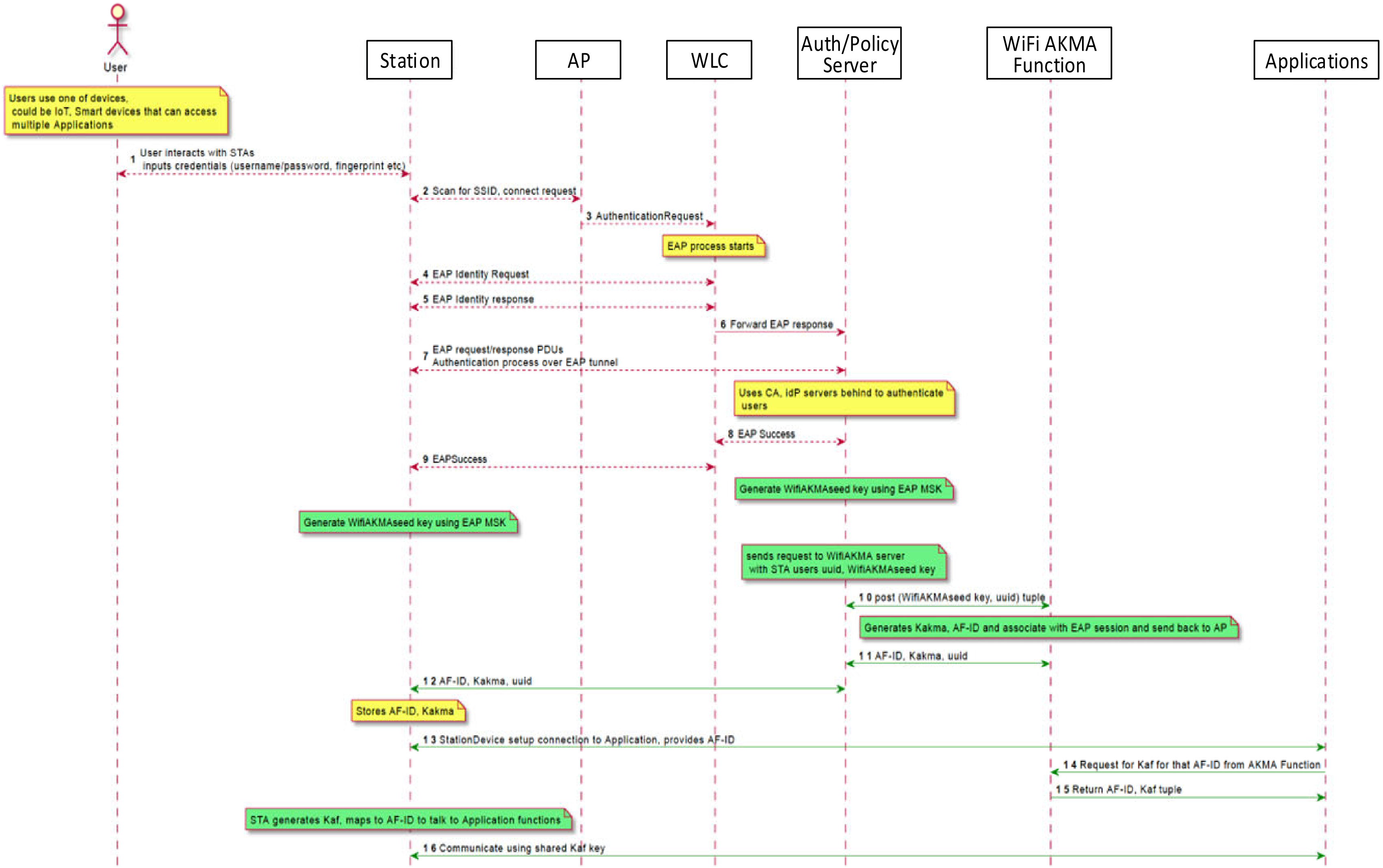
User
Station
AP
WLC
Auth/Policy Server
WiFi AKMA Function
Applications
Users use one of devices, could be IoT, Smart devices that can access multiple Applications
1 User interacts with STAs inputs credentials (username/password, fingerprint etc)
2 Scan for SSID, connect request
3 AuthenticationRequest
EAP process starts
4 EAP Identity Request
5 EAP Identity response
6 Forward EAP response
7 EAP request/response PDUs Authentication process over EAP tunnel
Uses CA, IdP servers behind to authenticate users
8 EAP Success
9 EAPSuccess
Generate WifiAKMAseed key using EAP MSK
Generate WifiAKMAseed key using EAP MSK
sends request to WifiAKMA server with STA users uuid, WifiAKMAseed key
1 0 post (WifiAKMAseed key, uuid) tuple
Generates Kakma, AF-ID and associate with EAP session and send back to AP
1 1 AF-ID, Kakma, uuid
1 2 AF-ID, Kakma, uuid
Stores AF-ID, Kakma
1 3 StationDevice setup connection to Application, provides AF-ID
1 4 Request for Kaf for that AF-ID from AKMA Function
1 5 Return AF-ID, Kaf tuple
STA generates Kaf, maps to AF-ID to talk to Application functions
1 6 Communicate using shared Kaf key

Figure 2: Illustrative Sequence Diagram

The sequence diagram that is presented in Figure 2, above, identifies a series of steps which are labeled 1 through 16 in the figure. Those steps, and in particular the activities that are highlighted in green in the figure, will be described below.

Broadly speaking, under a Step 0, authentication takes place as part of an “Association" step and a "Re-association" process. During Steps 1 and 2, an STA device discovers an AP using existing techniques. Under Steps 4 through 8, an AP, a wireless local area network (LAN) controller (WLC), and the auth/policy server employ an existing EAP protocol to authenticate between the AP and the ISE.

At Step 9, at the end of a successful authentication process (e.g., as described under Section 5.4 of the Internet Engineering Task Force (IETF) Request for Comments (RFC) 4851 for the EAP-FAST protocol) a Masterkey for that EAP session is generated from which a master SessionKey (MSK) and an extended master SessionKey (EMSK) may be generated. Importantly, from an MSK value, both the STA device and the auth/policy server may generate a key that will serve as the seed key for the AKMA keys. Such a seed key may be referred to as a WifiAKMASeedKey. Note that this key is equivalent to a KAUSF value in the above-described 3GPP AKMA approach.

Under Step 10, the auth/policy server posts the WifiAKMASeedKey and a universally unique identifier (UUID) tuple of the user and the STA device to a WiFi AKMA function. As noted previously, the WiFi AKMA function is a new logical function according to the techniques presented herein and in practice that function may be implemented by extending an auth/policy server or it may be a separate service that is invoked from an auth/policy server. In the sequence flow that was presented in Figure 2, above, it is shown as a separate function.

During Step 11, the WiFi AKMA function generates a KAKMA and AF-ID that are associated with the instant session and sends back to the auth/policy server the {AF-ID, KAKMA} tuple along with a unique identifier that identifies a user (e.g., a UUID as per RFC 4122). Such a UUID may be derived by a (e.g., ISE) server as part of an identity request/response step to uniquely identify that a user or device is connecting. Additionally, it is possible to obtain such a UUID through other means (if, for example, an enterprise already has a framework to anonymize a user’s real identity and employ UUIDs).

Under Step 12, the auth/policy server sends the {AF-ID, KAKMA, UUID} tuple to the STA device. Then at Step 13, the STA device connects to the applications where it wishes to send periodic data. After an existing secure connection channel is established, the STA device may present an AF-ID value to different applications. Under Steps 14 and 15, applications request the KAF key for the AF-ID from the WiFi AKMA function and that function returns the KAF key for the identified application group. Finally, at Step 16, applications and the STA device are able to communicate securely using the KAF key.

Use of the techniques presented herein offers a number of business values. With the proliferation of more and more IoT devices in an enterprise space, WiFi technology is increasingly becoming just as important as private 5G technology. Already, WiFi 6, 7, and 8 all contain a high focus on IoT use cases. The native support of application authentication with WiFi, as supported by the techniques presented herein, will have a long foreseeable benefit for many network equipment vendor product lines such as auth/policy servers, APs, and so on.

In summary, techniques have been presented that support a mechanism that extends the WiFi authentication process to support application server authentication for constrained devices. Aspects of the presented techniques support an exchange of a WiFi key and a key identifier (which may be referred to herein as a KAKMA key and an A-KID) as part of an EAP tunnel using a new IE once an authentication process has successfully completed. Such an exchange allows a STA device to use the key tuple {KAKMA, A-KID} to access any application functions that are grouped with that key identifier (i.e., A-KID) without requiring any further authentication.