March 2022

APPLICATION CENTRIC GROUPING USING AUTHENTICATION AND KEY MANAGEMENT FOR APPLICATIONS (AKMA) TECHNIQUES

Rajesh I. V

Ram Mohan R

APPLICATION CENTRIC GROUPING USING AUTHENTICATION AND KEY MANAGEMENT FOR APPLICATIONS (AKMA) TECHNIQUES

AUTHORS:
Rajesh I V
Ram Mohan R

ABSTRACT

The 3rd Generation Partnership Project (3GPP) Authentication and Key Management for Applications (AKMA) mechanism leverages an operator's authentication infrastructure to secure the communication between a User Equipment (UE) and an Application Function (AF). Techniques are presented herein that support establishing a grouping mechanism on top of the AKMA layer to efficiently group the applications and the AFs, using a limited set of keys, to simplify various use cases. The use of the techniques presented herein offers a number of optimizations and benefits, including when an AF needs to send a dynamic update to all the UEs that are managing specific applications, this can be efficiently done with a single group key; a UE need maintain only a single key to manage multiple applications from a same vendor; a grouping can be supported for a specific set of UEs, a specific region, or a specific time; etc.

DETAILED DESCRIPTION

As an initial matter, it will be helpful to confirm the meaning of a number of Authentication and Key Management for Applications (AKMA)-related terms that appear in the narrative that is presented below. Specifically:

Term	Meaning
A-KID	AKMA Key IDentifier of the User Equipment (UE).
AAnF	AKMA Anchor Function.
K_{AUSF}	AKMA AUthentication Server Function (AUSF) Key.
K_{AF}	AKMA Application Key.
K_{AKMA}	AKMA Anchor Key.
AF_ID	Application Function Identifier. These identifiers (IDs) are used in the K_{AF} derivation by both an Application Function (AF) and a UE. These IDs are known upfront to a UE and an AF as part of a registration process.

Published by Technical Disclosure Commons, 2022

AKMA is a mechanism that was introduced in the 3rd Generation Partnership Project (3GPP) technical specification (TS) 33.535 release 16, and further enhanced in release 17, that leverages an operator's authentication infrastructure to secure the communication between a UE and an AF. Apart from simplifying the bootstrapping of Internet of things (IoT) devices and saving the cost of credential management for application providers, the AKMA mechanism principally addresses the limitations of IoT devices in applying a username and password or certificate-based options during authentications. AKMA derives a final application key (i.e., K_{AF}) through a hierarchy of keys in a process which is eventually used between a UE and an AF. The application keys are derived on a per-application basis to avoid any hijacked application traffic influencing other application providers.

It is likely that a UE or a device will use multiple applications, with many of those applications belonging to the same vendor or domain. Additionally, there may be multiples of a specific type of a device which use similar applications and communicate with the same AF.

A grouping of a same vendor's applications in a UE and a grouping of UEs based on the same AFs would be useful for optimized communication for different use cases. Such groupings, according to aspects of the techniques presented herein, may encompass a number of use cases and benefits.

First, co-relating the data of multiple applications of the same UE, or an applications-based grouping of UEs, may be used in UE-based or application-centric analytics. Next, the processed data may be used in assessing the functional and health requirements of the UE and the applications. If an AF needs to send a dynamic update to the applications that are managed by it, this may be efficiently done with a single key rather than having multiple keys per application basis.

Further, a grouping may also be done selectively or be specific to a region in a dynamic and time-bounded manner. Constrained devices will not be required to preserve multiple keys when the applications are effectively grouped to handle a single key. Between AFs, data may be freely passed that is specific to the UEs in the group without managing multiple keys. Finally, it is possible to eliminate layers of handshaking with an

application (as will be discussed in the below narrative). However, AKMA does not expose any intelligence for the grouping of applications within a UE or across a UE that is specific to a vendor. AKMA creates a key on a per application basis for each UE, which is not efficient for the use cases that were presented above.

To address the types of challenges that were described above, techniques are presented herein that support a mechanism that builds a grouping capability on top of the AKMA framework using a limited set of keys.

As described above, the 3GPP's AKMA is a mechanism that leverages an operator's authentication infrastructure to secure the communication between a UE and an AF. Aspects of the techniques presented herein establish a grouping mechanism on top of the AKMA layer to efficiently group the applications and AFs, using a limited set of keys, to simplify various use cases.

Aspects of the techniques presented herein support two different grouping mechanisms. A first mechanism encompasses application grouping within a UE based on a same vendor AFs. A second mechanism encompasses application grouping across UE based on a same AF. Each of the mechanisms will be described and illustrated in the narrative that is presented below.

Before presenting the two grouping mechanisms, it will be helpful to first describe the AKMA subscription procedure.

According to the 3GPP standards, UEs requiring AKMA support will be subscribed accordingly. Those details are stored in a Unified Data Management (UDM) facility. Further, each AF will be identified with a unique AF_ID within a 5G System (5GS). As specified by the 3GPP, an AF_ID contains a fully qualified domain name (FQDN) which carries vendor domain details. Additionally, a same vendor may have multiple AFs.

In addition to the existing support, aspects of the techniques presented herein support an AKMA grouping policy which is specified by a vendor. The grouping policy captures the intent of using a grouping that is specific to AFs or at a UE level along with more granular instructions (such as, for example, if a policy needs to be applied at any specific region and time). Aspects of the presented techniques also provide support for changing the AKMA grouping policy on an on-demand basis by a vendor with a 5G

provider. While processing AKMA support of a UE at the time of authentication, the grouping policy will be supported as specified.

The next section of the narrative describes and illustrates the operation of the first grouping mechanism, which, as described above, encompasses application grouping within a UE based on a same vendor AFs.

During the primary authentication procedure, an AUSF interacts with the UDM in order to fetch authentication information such as subscription credentials (e.g., Authentication and Key Agreement (AKA) authentication vectors) and the authentication method.

In response, the UDM may also indicate to the AUSF whether the AKMA Anchor Keys need to be generated for the UE. Also, if an AKMA grouping policy has been specified then the UDM checks if the respective group was already created (e.g., is operational) and in the case where this is the first UE coming under that group then a new GROUP_ID will be created. Such a GROUP_ID will be uniquely identified within the respective 5GS and will be used during key generation. The UDM shares the GROUP_ID along with an AKMA indication. If a grouping policy was not specified and if AKMA is enabled then only the AKMA indication is sent to AUSF.

Under aspects of the techniques presented herein, a Vendor ID may be used as the GROUP_ID. The vendor identification may be the one that is present in an AF_ID as described above. If multiple vendor applications are used, then isolating the grouping policy may be done effectively with this approach.

If the AUSF receives the AKMA indication from the UDM, the AUSF stores the K_{AUSF} and generates the AKMA Anchor Key (K_{AKMA}) and the A-KID from the K_{AUSF} after the primary authentication procedure is successfully completed. The UE generates the AKMA Anchor Key (K_{AKMA}) and the A-KID from the K_{AUSF} before initiating communication with an AKMA AF. After the AKMA key material is generated, the AUSF selects the AAnF and sends the generated A-KID and K_{AKMA} key to the AAnF together with the Subscription Permanent Identifier (SUPI) of the UE and, if applicable, the GROUP_ID.

When the UE initiates communication with the AKMA AF, it includes the derived A-KID in the Application Session Establishment Request message. If the AF does not have

an active context associated with the A-KID, then the AF selects the AAnF and sends a request to the AAnF with the A-KID to request the K_{AF} key for the UE. The AF also includes its identity (i.e., an AF_ID) in the request.

The AAnF checks whether the AAnF can provide the service to the AF based on the configured local policy or based on the authorization information or the policy that is provided by the NF (Network Function) Repository Function (NRF) using the AF_ID. If successful, the procedures that are described below are executed. Otherwise, the AAnF rejects the procedure. The AAnF verifies whether the subscriber is authorized to use AKMA based on the presence of the UE-specific K_{AKMA} key that is identified by the A-KID. If a K_{AKMA} key is present in the AAnF, the AAnF derives the K_{AF} key from the K_{AKMA} key if it does not already have the K_{AF} key, using the GROUP_ID along with the K_{AKMA} key.

The AAnF sends the K_{AF} key and the K_{AF} expiration time to the AF. The expiration time is specified as the lifetime of a group as captured in a grouping policy. From the GROUP_ID the AAnF knows the vendor applications and AF work as a group. It generates a key with K_{AKMA} using the GROUP_ID. This remains the same for all of the applications of the vendor and mappings are captured. The AKMA context will capture the grouping information as well. The AF sends the Application Session Establishment Response to the UE. Afterwards, the UE creates a K_{AF} key with the K_{AKMA} key and the GROUP_ID.

If another application of the UE of a same vendor initiates a communication request with the respective AF (e.g., AF2), then AF2 approaches the AAnF as described above. The AAnF will determine that it has to be grouped and will share the same key to AF2.

As an optimization, the AAnF can proactively push the key to all of the AFs that are part of the group, or the AFs can periodically request the AAnF to share the created key, so that when the actual request comes from a UE it will be automatically honored. An AF does not need to communicate with the AAnF to obtain the key, thus optimizing the handshaking time.

Figure 1, below, depicts elements of a sequence diagram according to aspects of the techniques presented herein and reflective of the discussion of the first grouping mechanism that was presented above.

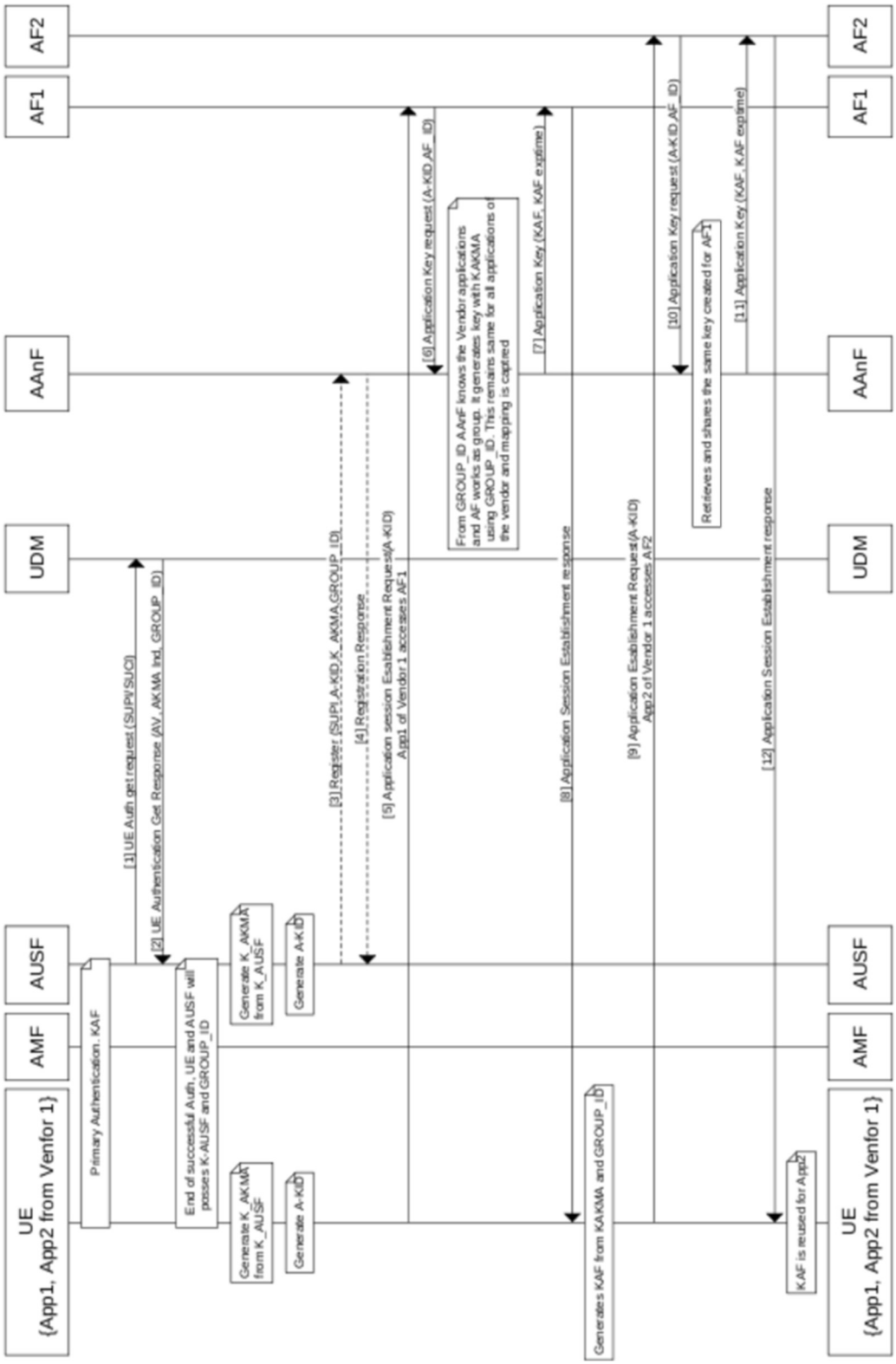

Figure 1: Application Grouping Within UE Based on Same Vendor AFs

The next section of the narrative describes and illustrates the operation of the second grouping mechanism, which, as described above, encompasses application grouping across UE based on a same AF.

Under this grouping mechanism, whenever the first UE successfully authenticates with a 5GS, based on the grouping policy the AUSF keys are generated. In this case a new key $K_{\_GAUSF}$ is derived which is used as the same base key for all of the UEs coming under the group. If the UE has a single application, then this key alone is sufficient. If it has more applications, then the regular key K_{AUSF} will also be created.

The GROUP_ID that is generated will be pointing to an AF_ID. Figure 2, below, depicts elements of a sequence diagram according to aspects of the techniques presented herein and reflective of the discussion of the second grouping mechanism that was presented above.

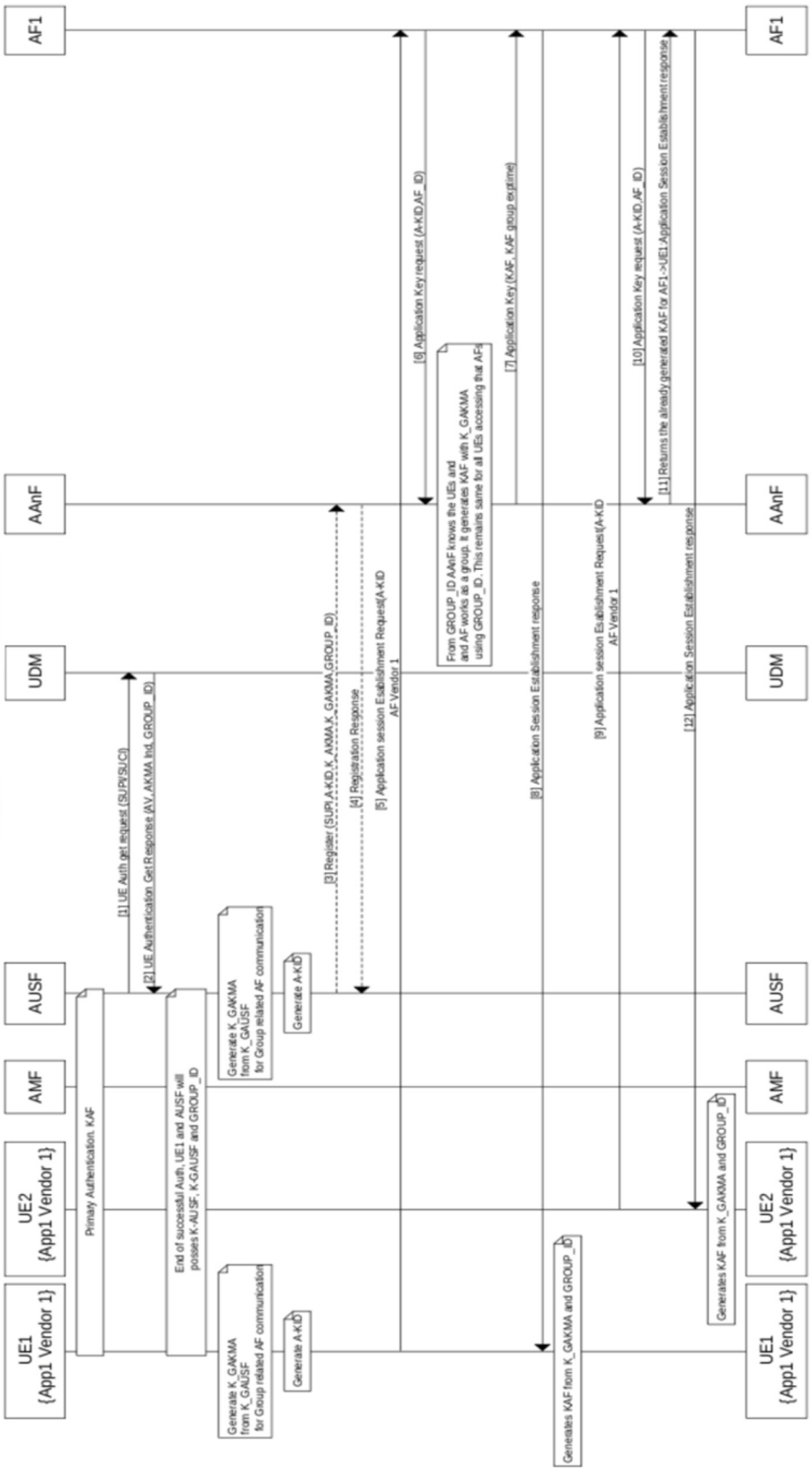

Figure 2: Application Grouping Across UEs Based on Same Vendor AFs

The use of the techniques presented herein offers a number of optimizations and benefits.

First, several of the optimizations and benefits may be described from an AF perspective. When an AF needs to send a dynamic update to all the UEs managing the specific applications, this can be efficiently done with the single group key. Otherwise, an AF needs to maintain different keys and encrypt the data that is specific to each UE differently. For any emergency data exchange this reduces latency and resource usage.

Further, for UE centered analytics different AFs serving the UE need to exchange data. With the group key, AFs can exchange the data of a UE without an explicit key transaction. Whenever the first application connects to the AF the group is created. The AKMA system can proactively send the group key to all of the concerning AFs. This will greatly reduce the second AFs undergoing the handshake procedure to obtain the key again for the subsequent applications.

Next, various of the optimizations and benefits may be described from a UE perspective. A UE can maintain only a single key to manage the multiple applications from a same vendor. This could be the likely case when multiple applications are used for a solution point of view.

Finally, several of the optimizations and benefits may be described from a system perspective. For example, a grouping can be made more granular and dynamic in nature. For example, a grouping can be supported for a specific set of UEs, a specific region, or a specific time. The AKMA grouping policy can be dynamically instructed by the user.

Further, the AKMA process is triggered once the primary authentication is completed and the lifetime ends when the UEs are out of network. With a grouping, even when a UE goes out of network, the group remains along with support for the rest of the devices. Importantly, a disconnected device may join the group any time. Additionally, there is flexibility to specify the lifetime of a group.

As described an illustrated in the narrative that was presented above, aspects of the techniques presented herein enable the ability to support application grouping within a UE based on the same vendor AFs on an AKMA system; the ability to support application grouping across UE based on a same AF on an AKMA system; the ability to specify the AKMA grouping policy in a dynamic manner; the ability to support granularity in grouping

such as a specific set of UEs, a specific region, or a specific time; and the ability to proactively send the keys to the concerning AFs for reducing key negotiation latency.

Additionally, for a network equipment vendor that supports private 5G and 5G-as-a-Service (5GaaS), AKMA grouping support may be offered as value-added service to support a customer's AFs and applications where, for example, a policy enforcement facility may be front-ended as a policy engine to handle AKMA policy and grouping.

In summary, techniques have been presented that support establishing a grouping mechanism on top of the AKMA layer to efficiently group the applications and the AFs, using a limited set of keys, to simplify various use cases. The use of the techniques presented herein offers a number of optimizations and benefits, including when an AF needs to send a dynamic update to all the UEs that are managing specific applications, this can be efficiently done with a single group key; a UE need maintain only a single key to manage multiple applications from a same vendor; a grouping can be supported for a specific set of UEs, a specific region, or a specific time; etc.