IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 29, 2017, accepted January 26, 2018, date of publication February 2, 2018, date of current version March 16, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2801383

A Lightweight and Flexible Encryption Scheme to Protect Sensitive Data in Smart Building Scenarios

SALVADOR PÉREZ[1], JOSÉ L. HERNÁNDEZ-RAMOS[1], SARA N. MATHEU-GARCÍA[1], DOMENICO ROTONDI[2], ANTONIO F. SKARMETA[1], LEONARDO STRANIERO[2], AND DIEGO PEDONE[2]

[1]Department of Information and Communication Engineering, University of Murcia, 30100 Murcia, Spain
[2]FINCONS SpA, 70124 Bari, Italy

Corresponding author: Salvador Pérez (salvador.p.f@um.es)

This work was supported in part by the Regione Puglia Avviso Aiuti a Sostegno dei Cluster Tecnologici Regionali per l'Innovazione within the Energy Router Project under Grant HX8HXI1, in part by the Spanish Ministry of Economy and Competitiveness through the CICYT EDISON project under Grant TIN2014-52099-R, and in part by the USEIT project under Grant CHIST-ERA PCIN-2016-010.

ABSTRACT Smart buildings represent key environments to encourage the growth of more sustainable and efficient cities. With the strong development of the Internet of Things (IoT), the integration of heterogeneous physical devices fosters the emergence of data-driven services to make more effective decisions accordingly. However, the need for sharing large amounts of data could help to infer users' sensitive information, such as their daily habits, thus harming their privacy. Under these premises, this paper introduces an encryption scheme based on the lightness of the symmetric cryptography, and the expressiveness of attribute-based encryption. Our proposal aims to ensure only authorised services will be able to access specific pieces of data, so that users' privacy is not compromised, while scalability and efficiency are provided. The resulting scheme has been deployed on a real smart building scenario, and validation results demonstrate its suitability to protect large amounts of sensitive data on IoT-enabled buildings.

INDEX TERMS Attribute-based encryption, confidentiality, key management, smart building, symmetric key cryptography.

I. INTRODUCTION

Smart Cities [1] have overwhelmingly emerged as the answer to cope with the demographic challenges associated with an increasingly urbanized population. As part of the integrative vision of smart cities, *buildings* represent a key environment for the development of more sustainable and efficient cities. In fact, according to the European Alliance of Companies for Energy Efficiency in Buildings (EuroACE [2]), we spend over 90% of our time in buildings. At the same time, we create 2.5 billion bytes of data each day [3]; consequently, most of the data we produce is originated within a building.

In this context, the so-called *Internet of Things* (IoT) [4] paradigm is considered as the main enabler to transform existing residential and industrial buildings to be "smart" [5], [6]. Smart buildings represent a heterogeneous ecosystem where different types of devices, such as Radio Frequency Identification (RFID) readers, Heating, Ventilating and Air Conditioning (HVAC) systems, or even legacy devices, generate large amounts of data related to their daily activity. These data are further processed by certain services in order to extract knowledge and make effective decisions regarding energy saving, emergencies and disaster management, or to provide a more habitable and comfortable environment. However, the development of these data-driven services has a dark side: users' privacy. Beyond obtaining individual data (e.g. an energy consumption measurement from a smart meter), the application of modern data aggregation and correlation techniques can help to infer users' daily habits or track them without their explicit consent. These data maximization trends takes users' privacy to a broader dimension, which may have safety implications if appropriate countermeasures are not implemented. Therefore, there is a real need to design approaches that ensure that this information is only accessible to authorized users or entities in order to protect users' privacy, while the functionality of such services can be still provided.

In order to address these challenges, this work proposes the use of a lightweight and flexible encryption scheme intended to protect sensitive data that are generated from different heterogeneous devices in an IoT environment. In this sense, we are based on our previous work [7] and we add the architecture definition of such scheme, its application on a real IoT-enabled scenario and the obtaining of results about its performance. Our proposal combines the lightness and efficiency of symmetric key cryptography [8] to protect data, with the expressiveness and flexibility of the *Ciphertext-Policy Attribute-Based Encryption* scheme (CP-ABE) [9] to distribute the corresponding symmetric keys. Indeed, we have also decided to combine the *Symmetric* and *CP-ABE* words to call the proposed scheme as SymCpAbe. The main aim of SymCpAbe is to provide a scalable solution that allows to protect huge amounts of data, while it does not require cumbersome key management and revocation tasks. Specifically, data from physical devices (e.g. a sensor) are protected by using a symmetric key algorithm, while CP-ABE encryption is delegated to an external service, in order to alleviate the burden of end devices. Thus, data are encrypted by using the AES algorithm with ephemeral symmetric keys, while these keys are protected through the CP-ABE scheme, so that users are able to maintain access control over their data, avoiding data leakages to unauthorized entities. Therefore, our approach facilitates the key management and distribution processes, ensuring a high level of scalability while improving the interoperability by using recent standards to represent the cryptographic material. Indeed, our approach makes use of *JSON Web Key* (JWK) [10] to represent the required cryptographic keys, and *JSON Web Algorithm* (JWA) [11]) to identify the corresponding algorithms. The resulting scheme has been deployed on a real smart building scenario to protect data from several devices (e.g. RFID readers, smart meters) that are shared to different high-level applications, such as the emergency services. The experimentation results and the security analysis demonstrate the feasibility of our solution, as well as the advantages from the application of an encryption approach based on symmetric key cryptography and CP-ABE schemes.

The remainder of the paper is structured as follows. Section II summarizes other works addressing data protection and sharing among groups of entities, taking into account aspects such as efficiency, flexibility or heterogeneity. In Section III, we provide an overview of the SymCpAbe scheme and how it addresses the highlighted challenges. Section IV shows the application of our solution into a real smart building scenario. Section V details the main interactions among the entities of SymCpAbe. Furthermore, Section VI shows a performance analysis of the proposed scheme, and Section VII discusses security and practical aspects related to our approach. Finally, Section VIII concludes the work and introduces some considerations on our future work in this area.

II. RELATED WORK

The emerging IoT scenarios envisage groups of heterogeneous devices exchanging a significant amount of information through cloud platforms over loosely coupled interaction patterns. Such interactions are usually based on short-lived communications, so security models enabling to protect this information must take into account this aspect, that is, asynchronous communications, beyond typical multicast solutions [12]–[14]. In order to address theses issues, Attribute-Based Encryption (ABE) schemes [15] are receiving increasing attention due to its high level of flexibility and expressiveness, compared to traditional symmetric and public key approaches. In ABE schemes, entities are represented by identity attributes, so data will be accessible only to participants satisfying specific combinations or sets of attributes. Based on ABE, CP-ABE [9] allows data to be encrypted under a logical combination of identity attributes (access policy), while private keys are associated with a set of attributes. Therefore, data will be decrypted only by those entities whose private keys satisfy the conditions specified in the access policy.

The application of the CP-ABE scheme to provide confidentiality on group data sharing scenarios has been considered in other scenarios, such as in *E-Heatlh* [16], [17], *Financial Industry* [18] or *Social Networks* [19], where entities leverage the flexibility of access policies to guarantee that information will only be accessed by authorized users. In addition, the information is typically shared through cloud platforms following the *publish/subscribe pattern*, in order to enable entities to be decoupled from each other. It should be pointed out that these proposals assume that devices are able to successfully execute the resource-demanding CP-ABE encryption operations. Regarding this assumption, the scientific literature reports studies that examine the feasibility of using this scheme on devices with different features. In particular, [20] presents a performance analysis in which the execution of CP-ABE both on a common laptop and a smartphone is compared, demonstrating that the computer achieves acceptable results, while the use of this scheme on current smartphones or similar devices is still challenging when a high security level is required (i.e. 112-bits or higher [21]). In contrast to this work, the authors of [22] provide CP-ABE implementation (*ANDRABEN*)[1] based on [51], and analyse its application on current smartphones, proving that a reasonable performance in these devices can be achieved. However, they do not provide scalability tests considering huge amounts of information coming from different devices need to be protected. Towards this end, SymCpAbe has been deployed and tested on devices with similar hardware to smartphones that use the CP-ABE scheme to protect data. In addition, unlike the previous proposals, we have also considered these devices have to manage several incoming data

[1] http://spritz.math.unipd.it/projects/andraben/

received from different data sources with the purpose to check and assure the scalability of our solution.

Furthermore, in our solution, physical devices delegate the CP-ABE encryption operation to an external service, in order to alleviate their burden. Indeed, the expensiveness of CP-ABE cryptographic operations has motivated the emergence of different solutions in which such operations are delegated to more powerful entities. Touati *et al.* [23] present a solution where resource-constrained devices assign the CP-ABE operations to more powerful devices, assuming that these are trusted. Thus, information is sent from data sources to these assistant entities, which encrypt it by using the CP-ABE scheme. The result is returned to the originating entities, or forwarded to a central platform. Nevertheless, note that all information is sent to assistant entities without protection, so that if these are compromised, an attacker could access the data. Similarly, [24] defines a scheme that extends the CP-ABE approach and allows entities to delegate the most overhead of decryption operations to a cloud platform. In addition, this solution adds a *Message Authentication Code* (MAC) to each ciphertext with the purpose to verify whether the computed result is correct, thus preventing potential counterfeits. Despite this, the proposed scheme has not been tested in scenarios where devices need to handle a high amount of incoming data.

As already mentioned, the proposed approach combines the efficiency of symmetric key cryptography with the flexibility of the CP-ABE scheme. In this direction, [25] describes a solution in which information is encrypted by using the AES algorithm with symmetric keys, which are, in turn, protected under the CP-ABE scheme. Subsequently, AES protected data along with their CP-ABE encrypted associated keys are stored on the cloud. Therefore, only those entities whose CP-ABE private key satisfy the access policy used to encrypt the symmetric key, will be able to decrypt the information. Likewise, [26] describes a new approach in which data are shared among several entities following the *publish/subscribe pattern*. This proposal uses the AES algorithm to protect the information to be shared with groups of devices, which are managed by a controller entity. Additionally, the CP-ABE scheme is used to protect symmetric keys, generating key-update messages every time a new AES key is computed. However, although these solutions combine both cryptographic schemes, they do not take into account scenarios where a huge volume of data has to be shared, so that scalability could be affected.

These research proposals partially address some of the main security challenges that arise in IoT scenarios. However, they do not take into account those scenarios made up by several devices exchanged large amount of information from each other, so that these proposals do not guarantee scalability. In this sense, our proposal is conceived to be deployed on this type of scenarios. Furthermore, in order to offer a more interoperable approach, SymCpAbe is based on recent security standards to represent the cryptographic algorithms and keys that are required to realize the intended functionality. In addition, the resulting scheme has been deployed and tested on a real smart building scenario that is described in the next sections.

III. OVERVIEW

The current trend towards a hyper-connected world makes users' basic privacy principles more difficult to be enforced. Given the scale and heterogeneity of potential IoT-enabled environments, data protection mechanisms must offer a high level of efficiency and flexibility to be accommodated on different devices with the purpose of preserving scalability. As part of these mechanisms, encryption algorithms represent an essential component to ensure only legitimate and authorised entities will be able to access the data. In this sense, beyond the use of traditional approaches based on symmetric and public key cryptography, emerging cryptographic schemes are being proposed as alternatives to be used in IoT scenarios [27]. In particular, the CP-ABE scheme has been widely used in recent works [20], [28], [29] in order to enable a scalable data protection mechanism for the IoT. However, as already mentioned, it requires extensive processing capabilities to execute highly resource-demanding cryptographic operations. This is especially relevant in scenarios where huge amounts of data will be exchanged, since scalability can be conditioned. Furthermore, end-to-end confidentiality can be compromised if encryption/decryption algorithms cannot be accommodated on end-devices, thus limiting the applicability.

In order to address such practical issues while the advantages of CP-ABE can be still leveraged, our approach combines symmetric key cryptography and CP-ABE, with the purpose of achieving a trade-off between scalability and efficiency. In particular, symmetric keys are used to protect data, whereas these keys are CP-ABE encrypted by using data owner's CP-ABE policies. Accordingly, an entity needs to get access the symmetric key to be able to access data. Then, if its CP-ABE key satisfies the CP-ABE policy, the entity will obtain the symmetric key and, consequently, the encrypted data. It should be pointed out that, in order to leverage the SymCpAbe efficiency, end-devices encrypt data by using symmetric key encryption, while CP-ABE encryption operations are delegated to an external service, so performance drawbacks related to the use of CP-ABE on end-devices are mitigated. In this sense, Figure 1 shows a high-level view of the proposed architecture for our approach, in which we have identified five main entities:

- ***CP-ABE Delegator***. This entity delegates CP-ABE encryption operations to the *CP-ABE Assistant* in order to protect symmetric keys. Such keys are employed to protect data that are included in events to be used by external services.
- ***CP-ABE Assistant***. It performs the CP-ABE encryption process to protect symmetric keys, which are stored in the *KSS*.
- ***Key Storage Service (KSS)***. This service stores the CP-ABE encrypted symmetric keys, in such a way that

they can be obtained by *Applications* interested on different types of data.

- ***Event Storage Service (ESS)***. This service is in charge of storing events containing encrypted data from the *CP-ABE Delegator* to be provided to *Applications*.
- ***Applications***. They represent entities interested on receiving events, in order to provide data-driven services.

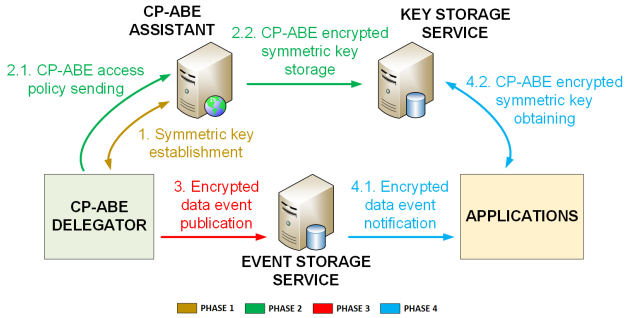

FIGURE 1. SymCpAbe architecture and interactions overview.

Furthermore, Figure 1 shows the main phases we have established for the SymCpAbe approach. Specifically, the first phase (*Symmetric key establishment - step 1*) is focused on setting up a new symmetric key that will be used by the *CP-ABE Delegator* to protect certain data. The second phase (*Symmetric key encryption and storage - steps 2.1 and 2.2*) encompasses the encryption and storage of the symmetric key. Even though the CP-ABE encryption process is delegated to the *CP-ABE Assistant*, the *CP-ABE Delegator* is intended to maintain the control on how data are disseminated. Indeed, this entity represents the point where data owners are in charge of selecting the CP-ABE policy that will be used for encrypting the symmetric key, as well as to provide such policy to the *CP-ABE Assistant*. Once the symmetric key is encrypted with CP-ABE, it is stored on the KSS to be accessible for *Applications*. The third phase (*Encrypted data event publication - step 3*) focuses on the creation and publication of events including data encrypted with the symmetric key. It should be noted that such operation does not require the involvement of any third entity; the *CP-ABE Delegator* directly publishes encrypted data events on the *ESS*. This way, data are end-to-end protected. The last phase (*Encrypted data event retrieval - steps 4.1 and 4.2*) embraces event notifications to interested *Applications* and the retrieval of encrypted data included in such events. This way, encrypted data are only accessible to those *Applications* whose CP-ABE key satisfies the CP-ABE policy that was used for encrypting the corresponding symmetric key associated to such data.

The previous description aimed to provide an overview about the main components and interactions required for our approach. Next section describes the application of SymCpAbe on a smart building use case in order to highlight its suitability and applicability on a real IoT scenario.

IV. A SMART BUILDING USE CASE

Smart buildings represent a suitable scenario to demonstrate the applicability of the proposed encryption scheme. In these environments, large amounts of data coming from different heterogeneous devices need to be shared to enable data-driven services to make decisions accordingly; due to the amount and sensitivity of such information, users' privacy can be compromised if data protection mechanisms are not implemented. Specifically, the considered use case is based on the description proposed in [30], where an IoT-enabled smart building is presented to address energy efficiency aspects. Figure 2 shows a simplified overview of this use case, in which the SymCpAbe entities are integrated.

The scenario is represented by a real smart building where a set of devices, such as smart meters, fire detectors, RFID readers, and other sensors/actuators (e.g. smart door locks) are physically deployed. These appliances, including legacy devices, act as *Data Sources* and they are in charge of capturing data associated with the daily activity in the building. Then, this information is sent to *Gateway* devices, which represent a central point for homogenizing the data communication from heterogeneous data sources to the *Smart Building Platform*. This platform represents the set of services and components that are intended to enable an efficient exchange of huge amount of information from data sources to high-level *Services* (*Applications*) through the *Publish/Subscribe Broker*. Note that, while data sources and *Gateways* are located in the building, the platform's services can be deployed in the cloud, if necessary.

All data sources are registered in the *Resource Directory* component that stores information from each device (e.g. its location), to ensure that only previously registered devices are able to publish data on the platform. In addition, all users of the building and services interested in building data are registered in the *Identity Manager* component, which stores their associated identity attributes. Users hold an RFID card that unequivocally identifies them when accessing the building's facilities. Therefore, the *Identity Manager* stores their personal information (e.g. name, identifier or role) along with other data, such as mobility condition, which are associated with their corresponding RFID card number. Furthermore, identity attributes associated to *Services* are employed to generate the corresponding CP-ABE keys linked their specific sets of attributes. These keys are used to get access to data generated in the building.

Services are subscribed on the *Publish/Subscribe Broker* to be notified about information in which they are interested. For the sake of clarity, we have selected a small number of such services. Nevertheless, many other applications and devices could require to access specific information to provide a certain functionality. In this scenario, the building administrator and the utilities company are subscribed to smart meters' energy consumption data, in order to monitor building energy use and generate a proper energy plan accordingly. In case of fire detectors' data, the building administrator and the emergencies services are responsible for managing a

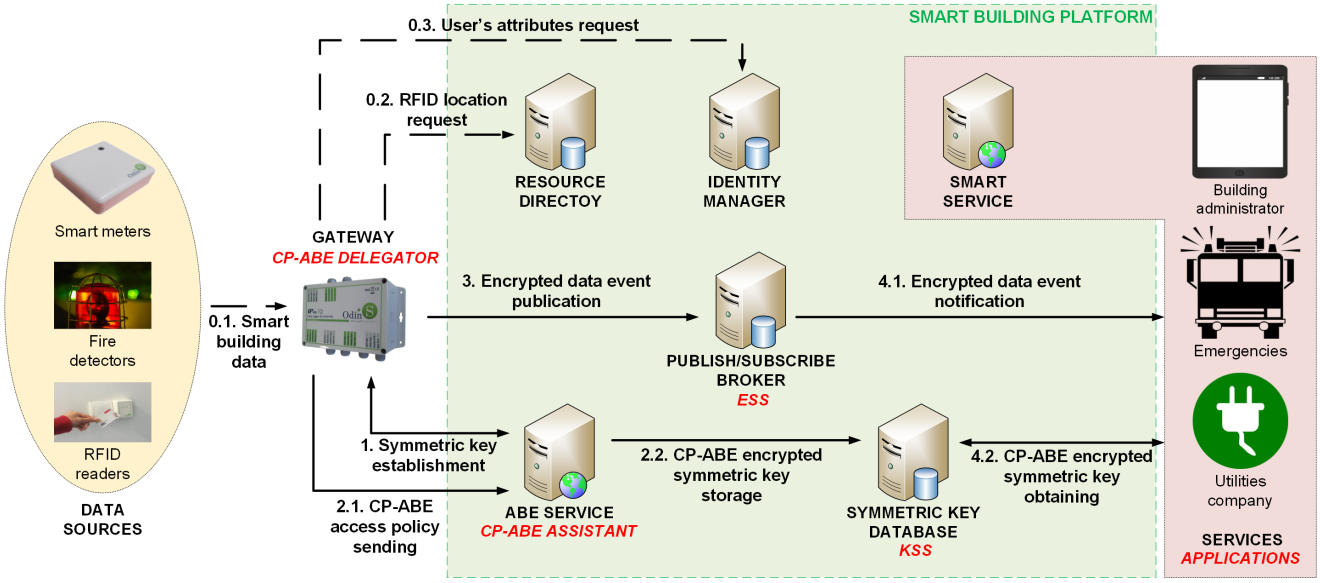

FIGURE 2. Application of our SymCpAbe scheme on a smart building use case.

potential critical situation based on received data. In addition to *external* services (e.g. the emergencies service), the *Smart Service* represents the *brain* of the building and it is part of the *Smart Building Platform*. This service is responsible of optimizing its electric consumption by making decisions such as turning on/off the lights or the HVAC system depending on the number of users in a certain room.

In addition, this service is in charge of detecting any suspicious behavior of users in order to anticipate potential eventual incidents; for example, to detect if unauthorized users attempt to perform illegitimate activities on a building facility. For this purpose, RFID readings are correlated with presence sensors' data to identify the position of each user at any time. Consequently, this service must be able to access all the information that is produced in the building, including sensitive users' data (for example, their position). This situation represents a typical case of trade-off between safety and privacy; while this information is useful for detecting anomalous situations in the building, such data must be properly protected so that no external or unintended services can track users without their consent. As an example, in a fire situation, some of users' data, such as their location and mobility condition are valuable to the emergency staff, since they will be able to define the most appropriate strategy for the building evacuation, prioritizing those places where there are users with restricted mobility. However, since users' location is inferred from RFID readings, the emergency service will also be able to access each user's identity, which may harm her privacy. By considering Figure 2, the use of SymCpABE is intended to efficiently and effectively mitigate the risks associated with the unintended disclosure of sensitive data.

Under these premises, when a *Gateway* receives data from *Data Sources* (*step 0.1*), it contacts the *ABE Service* (*ABES*) to establish a symmetric key (*step 1*). Such key has an associated lifetime, so that the *Gateway* will use it to protect incoming data of the same type (e.g. RFID reading) as long as the key is not expired; in case of key's expiry, a new key will be established. Then, the *Gateway* acts as *CP-ABE Delegator* and provides a CP-ABE access policy to the *ABES* (playing the *CP-ABE Assistant* role), in order to protect such key (*step 2.1*). In case of RFID readings, the Gateway sends the policy (role = "building_administrator or role = "smart_service"}, so only the building administrator and the smart service will be able to access such data. Subsequently, when the *ABES* receives the corresponding access policy, it encrypts the symmetric key by the CP-ABE scheme and stores it on the *Symmetric Key Database*, acting as the *KSS* (*step 2.2*).

It should be pointed out that previous interactions only will be performed in case a new symmetric key needs to be established; otherwise, the *Gateway* encrypts incoming data by using the established symmetric key, and generates a new event including such encrypted information. In particular, when data come from a RFID reading, it obtains the RFID readers location (*step 0.2*), as well as the user's identifier and mobility condition from the *Identity Manager* (*step 0.3*). Then, the *Gateway* only encrypts the user's identifier, which is included with the RFID reader's location and the user's mobility in the event. This way, these unprotected data are used by the emergency service, while the user's identifier is kept protected, this preserving users' privacy (this service does not know who user is). Once the event has been generated, it is published on the *Publish/Subscribe Broker* (as *ESS* (*step 3*)) that forwards it to those *Services* previously subscribed on such type of events (*step 4.1*). Then, *Services* request the CP-ABE encrypted symmetric key from the *Symmetric Key Database* (*step 4.2*) and try to decrypt it with their CP-ABE private keys. If the decryption process is successful,

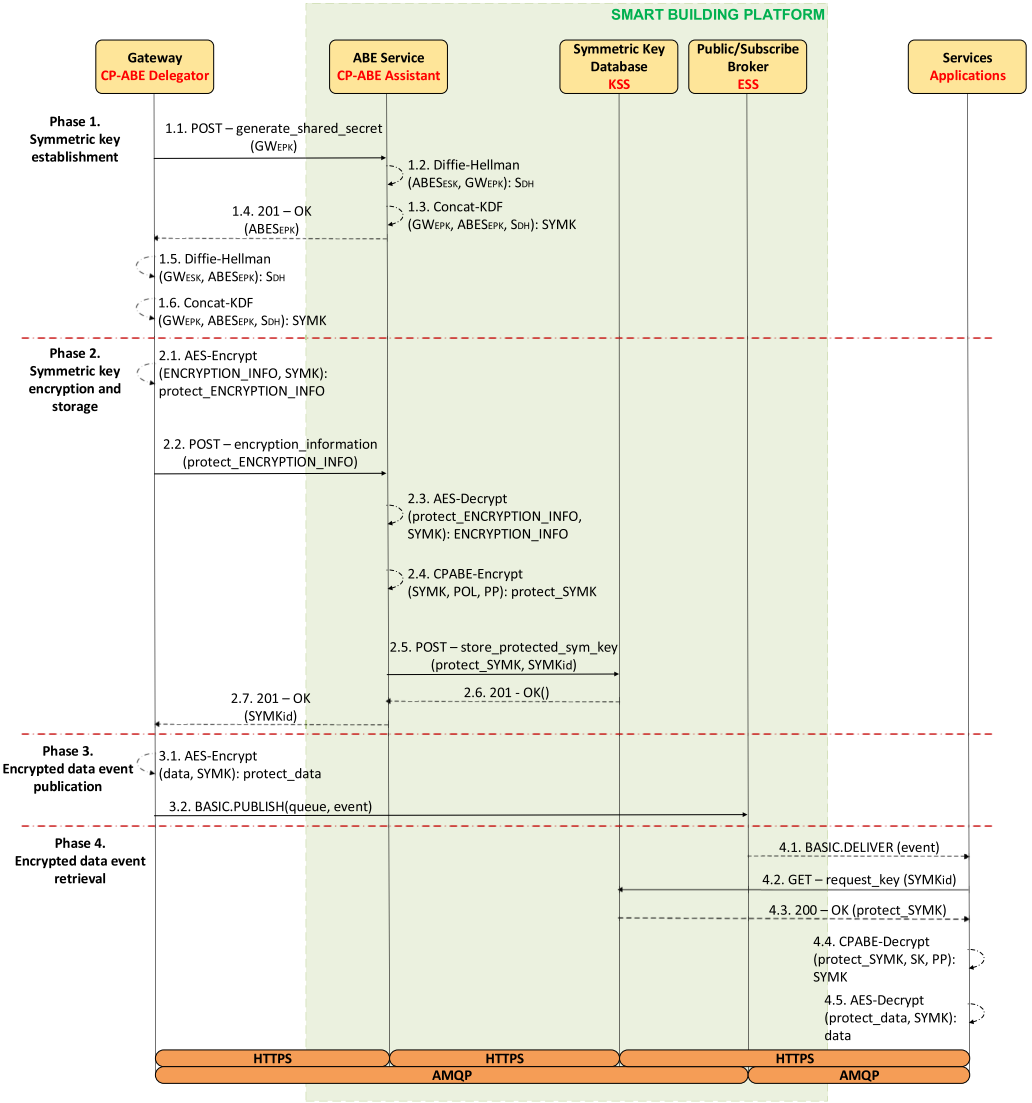

FIGURE 3. SymCpAbe interactions for the smart building use case.

the corresponding service will be able to retrieve data by using the decrypted symmetric key. In this case, while RFID reader's location and user's mobility will be accessible for all subscribed services, only the building administrator and the smart service will be able to access the user's identifier.

This smart building use case represents a real IoT-enabled scenario, where data coming from many heterogeneous devices need to be properly protected. In this sense, SymCpAbe has been proposed to supply such functionality, providing an efficient and flexible scheme while scalability is still preserved. Next section provides a detailed description of the main interactions required for the application of our scheme to the proposed scenario.

V. INTERACTIONS

In this section, we delve into the interactions performed by the entities of the use case previously described. Hence, Figure 3 shows a sequence diagram focused on the integration of SymCpAbe to the use case presented in previous section, identifying the required messages and processes at each phase. Note that, while HTTP and the *Advanced Message Queuing Protocol* (AMQP) [31] have been employed as application-layer protocols, other underlying technologies could be adopted, such as the Constrained Application Protocol (CoAP) [32] and the Message Queue Telemetry Transport (MQTT) [33]. Additionally, HTTP interactions are protected by using Transport Layer Security (TLS) with certificate-based mutual authentication [34]. Therefore, the messages related to key generation and distribution are protected. It should be noted that the data sharing approach by using AMQP follows a data-centric approach in which messages are not protected (e.g. through TLS), but data themselves. This way, the same single message can be used by the ESS to share a specific piece of data with a potential group of

potential services. Moreover, while not shown in the figure, IP-enabled *Data Sources* send their data by using CoAP messages that are protected with Datagram Transport Layer Security (DTLS) based on the Pre-Shared Key mode [35]. In case of legacy devices, this communication is carried out by using proprietary protocols. It should be pointed out that, for the sake of clarity, the description is focused on the components that are required to carry out the functionality of our scheme within the use case.

A. PHASE 0 (INITIAL CONFIGURATIONS)

In this preliminary phase, we assume that the *ABES* and *Services* obtain the CP-ABE public parameters (PP), in order to perform the cryptographic operations defined by the CP-ABE scheme. Similarly, *Services* get their corresponding CP-ABE private keys (SKs), associated with their set of attributes. In this sense, it should be pointed out that both processes are carried out through communication with an *Attribute Authority* entity, as described in [9] and [15]. On the other hand, we also consider that *Services* subscribe to the *Publish/Subscribe Broker* to be notified about any event referring to data in which they are interested.

B. PHASE 1 (SYMMETRIC KEY ESTABLISHMENT)

During this initial phase, the *Gateway* and the *ABES* establish a symmetric key (SYMK). For this purpose, the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) algorithm [36] is used. By using the ephemeral version of DH, the establishment of a new symmetric key will require a new key pair, thereby increasing the untraceability of the encrypted data flow. Thus, the *Gateway* firstly generates an ephemeral elliptic curve key pair by using a specific elliptic curve (e.g. NIST P-256). Then, it includes the public key and the selected curve into a GW_{EPK} structure, which is sent to the *ABES* (*step 1.1*). Listing 1 shows a GW_{EPK} example following the format specified by JWA [11]. Such information will be used by the *ABES* to set the parameters to be used for the ECDHE algorithm.

```
{
 "alg": "ECDH-ES",
 "enc": "A128GCM",
 "apu": "QWxpY2U",
 "apv": "Qm9i",
 "epk":
  {
   "kty": "EC",
   "crv": "P-256",
   "x": "gI0GAILBdu7T53akrFmMyGcsF3n5dO7MmwNBHKW5SV0",
   "y": "SLW_xSffzlPWrHEVI30DHM_4egVwt3NQqeUD7nMFpps"
  }
}
```

Listing 1. Example of ephemeral public key information.

- *alg* indicates the algorithm to generate SYMK (ECDHE).
- *enc* specifies the algorithm that will be used to encrypt data (AES GCM with a 128-bit key).
- *apu* and *apv* contain the *Gateway* and *ABES* identifiers, encoded as a base64url string.
- *epk* is the ephemeral public key represented as a JWK [10].
  - *kty* identifies the key type (EC).
  - *crv* that specifies the elliptic curve (P-256)
  - *x* and *y* parameters contain the EC point coordinates encoded in base64url.

When the *ABES* receives this message, it generates its own ephemeral elliptic curve key pair according to the value of *crv*. Then, it runs the ECDHE algorithm to calculate a shared secret (S_{DH}) with the *Gateway* (*step 1.2*) by using its ephemeral private key ($ABES_{ESK}$) and the GW_{EPK}. To enhance the strength of the shared symmetric key generation process, we have adopted the Concatenation Key Derivation Function (Concat-KDF) [37] to derive the SYMK from the S_{DH}. This function uses the GW_{EPK}, the *ABES* ephemeral public key ($ABES_{EPK}$) and the S_{DH} to generate SYMK (*step 1.3*). Then, the *ABES* sends the $ABES_{EPK}$ to the *Gateway*, following the example of Listing 1 (*step 1.4*). Upon receiving this message, the *Gateway* completes the ECDHE algorithm execution to obtain the S_{DH} (*step 1.5*) by using the $ABES_{EPK}$ and its ephemeral private key (GW_{ESK}). In addition, it executes the Concat-KDF function to derive the SYMK that will be shared by both entities (*step 1.6*).

C. PHASE 2 (SYMMETRIC KEY ENCRYPTION AND STORAGE)

This phase focuses on protecting the computed SYMK by using a CP-ABE policy (POL). Towards this end, the *Gateway* includes POL into a ENCRYPTION_INFO structure. Listing 2 shows an example of this structure.

```
{
 "timestamp": "2017-04-03T16:18:02Z",
 "device_id": "http://SmartBuilding/Gateway01",
 "policy":
  {
   "specs": "building_administrator or emergencies",
   "metadata": [
    {
     "name": "CreationDate",
     "value": "2017-03-24T12:34:32Z",
     "type": "http://sensorml.com/ont/swe/property/
          DateTimeStamp"
    }
   ]
  }
}
```

Listing 2. Example of information related to symmetric key encryption.

- *timestamp* indicates when the message was generated according to ISO 8601 [38] format. By following this format, the interoperability between the *Gateway* and the *ABES* is facilitated.
- *device_id* identifies the *Gateway*.
- *policy* provides details about POL to be used to encrypt the SYMK.
  - *specs* represents the POL as a tree data structure, where leaf nodes correspond to the different attributes and intermediate nodes are the AND/OR

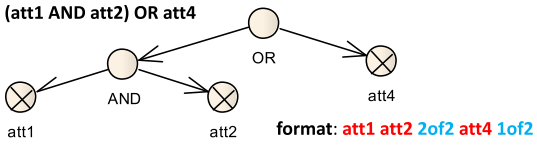

FIGURE 4. A CP-ABE access policy representation.

logical operators. A POL example is shown in Figure 4.

- *metadata* are a set of attributes providing additional information about the POL.

The ENCRYPTION_INFO is encrypted by using AES with the SYMK (*step 2.1*) and sent to the *ABES* (*step 2.2*). Upon receiving this message, the *ABES* decrypts the ENCRYPTION_INFO (*step 2.3*). Then, it executes the CP-ABE encryption operation with the provided POL to protect the SYMK (**2.4**). Furthermore, the *ABES* generates a unique key identifier associated with the protected SYMK ($SYMK_{id}$) that will be used by *Services* to get such SYMK at Phase 4. Next, the *ABES* stores the protected SYMK and the $SYMK_{id}$ on the *Symmetric Key Database* (*steps 2.5 and 2.6*). Then, the *ABES* sends the $SYMK_{id}$ to the *Gateway* (*step 2.7*). This identifier is used by the latter at Phase 3 to identify the SYMK that is employed to encrypt data of events. In addition, the *Gateway* establishes a limited lifetime for the SYMK ($SYMK_{lifetime}$). This way, when such $SYMK_{lifetime}$ expires, Phase 1 should be performed again. Therefore, in case SYMK is obtained by an attacker, it will only be able to recover the data encrypted with such specific key. Furthermore, note that the $SYMK_{lifetime}$ is based on the number of published events in order to delimit the amount of data that could be accessed in an unauthorized way, regardless of the *Gateway* publication rate.

D. PHASE 3 (ENCRYPTED DATA EVENT PUBLICATION)

At this stage, upon receiving incoming data, the *Gateway* uses the SYMK to encrypt them (*step 3.1*). Then, it creates a new event including the protected data along with the $SYMK_{id}$, the *Gateway* identifier and a set of metadata. Note that we have defined an event as a structure that follows the format specified in Listing 3.

```
{
 "device_id": "http://SmartBuilding/Gateway01",
 "symmetric_key_id": "541594b1-2f8d-431a-a5a4-666393
      e4adc4",
 "encrypted_data": "Ewhbw9e2cpyGaa5XDdOUoA==",
 "metadata": [
  {
   "name": "Description",
   "value": "Fire alarm",
   "type": "urn:org-emergencies:fire"
  }
 ]
}
```

Listing 3. Event example with encrypted data related to fire alarm.

- *device_id* is a URI that identifies the *Gateway*.
- *symmetric_key_id* unequivocally identifies the SYMK. This identifier is used by *Services* to retrieve such key from the *Symmetric Key Database*.
- *encrypted_data* contains the AES encrypted data as a base64url string.
- *metadata* are a set of attributes providing additional information about the data, such as the data creation date or the description.

Then, when the event is created, the *Gateway* publishes it on the *Publish/Subscribe Broker* by a BASIC.PUBLISH AMQP message, indicating a determined queue in which it will be stored (*step 3.2*). It should be pointed that aspects related to the AMQP are outside the scope of this work.

E. PHASE 4 (ENCRYPTED DATA EVENT RETRIEVAL)

This phase begins when a *Service* receives events from the *Publish/Subscribe Broker* through a BASIC.DELIVER AMQP message (*step 4.1*). Then, it performs a request to the *Symmetric Key Databases* with the $SYMK_{id}$ included in the received event to get the corresponding protected SYMK (*steps 4.2 and 4.3*). At this point, the *Service* tries to decrypt such SYMK using its SK previously obtained. If its SK satisfies the POL that was used to encrypt the SYMK, this *Service* will be able to decrypt it (*step 4.4*) and, therefore, this will be able to retrieve the encrypted data of the event by using the AES algorithm (*step 4.5*).

The above description is aimed to provide a comprehensive view of the SymCpAbe approach. Next section provides a detailed performance analysis that is intended to highlight the benefits of our approach in terms of efficiency, flexibility and scalability.

VI. PERFORMANCE ANALYSIS

The aim of this section is to demonstrate the advantages of the SymCpAbe approach in terms of performance. Towards this end, we compare our scheme with the direct application of CP-ABE to protect large amounts of data. It should be pointed out that, in current CP-ABE schemes and implementations, each piece of data is protected by using a *one time symmetric key*, which is in turn encrypted with CP-ABE. Thus, each ciphertext includes both the encrypted data and the corresponding CP-ABE encrypted symmetric key. Consequently, unlike our approach, the distribution of the symmetric key is not required. For comparison purposes, we have considered this approach, since it is widely adopted in current works, such as [20], [28], and [29].

According to the main entities that were identified in Section IV, Table 1 shows the hardware components and software libraries that we have used for evaluation purposes. Note that the proposed evaluation has been performed by considering different practical aspects, such as runtime, memory consumption, as well as the number of published events and attributes of the access policy (POL).

A. DATA EVENT PUBLICATION PERFORMANCE

This stage comprises the set of steps and operations required to protect and send the data to the *Publish/Subscribe Broker*.

TABLE 1. Features of devices employed in the smart building use case.

ENTITY	DEVICE (CPU)	RAM	SOFTWARE
Gateway	ARM Cortex-A53 (1.2 GHz)	1 GB	-
ABES	Intel Core i5 (2,7 GHz)	8 GB	CP-ABE library [39]
Symmetric Key Database	Intel Xeon E5-2660 (2,2 GHz)	7 GB	OrientDB 2.2.15
Publish/Subscribe Broker	Intel Xeon E5-2660 (2,2 GHz)	7 GB	RabbitMQ 3.6.5
Services	Intel Core i5 (2,7 GHz)	8 GB	CP-ABE library [39]

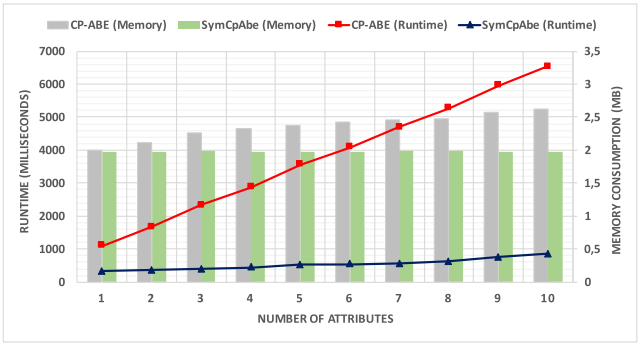

FIGURE 5. Memory and runtime for CP-ABE and SymCpAbe by the *Gateway*.

Specifically, it covers Phases 1-3 in the case of SymCpAbe, while for CP-ABE, the *Gateway* is responsible for encrypting each data by using such encryption scheme. It should be noted that SymCpAbe results have been obtained by considering "$SYMK_{lifetime} = 1$ event". Therefore, this can be considered as the *"worst case"* for our approach since Phases 1 and 2 must be performed every time a data is received by the *Gateway*. Thus, Figure 5 shows the memory and the average runtime required by the *Gateway* to publish a new encrypted data event by using both approaches. As shown, while the memory consumption increases according to the number of attributes in POL for CP-ABE, it remains constant under our approach since the CP-ABE encryption operation is delegated to the *ABES* (Phase 2). For the required runtime, in case of CP-ABE, it increases linearly (from 1105 ms for 1 attribute to 6525 ms for a 10-attribute POL). Note that even for the SymCpAbe *"worst case"*, the required runtime grows very slowly (from 328 ms to 857 ms for 1 and a 10-attribute POL, respectively), regardless the number of attributes in POL. Furthermore, it should be pointed out that if "$SYMK_{lifetime} > 1$ even", the runtime would be decreased, since the most time-consuming phase (Phase 2) is only executed when $SYMK_{lifetime}$ expires.

Moreover, Figure 6 shows a relative comparison regarding the percentage of published events by the *Gateway* (Phase 3) according to the incoming data rate, that is, how often new data are received. According to it, with a 0.25 data/second rate, the relative percentage of published events for both

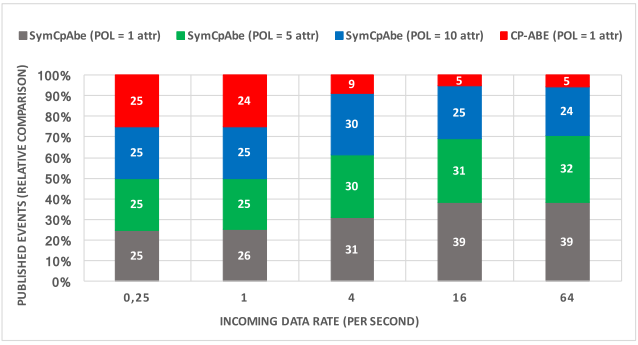

FIGURE 6. Percentage of published events by the *Gateway*, varying the data rate.

approaches remains similar. However, with a 4 data/second rate, this percentage is drastically reduced (from 25% to 9%) even when the most CP-ABE lightweight case (i.e. encrypting data with a 1-attribute POL) is used. The reason is that, with such rate, the *Gateway* is overloaded and it is not able to perform the resource-demanding CP-ABE encryption and publish all incoming data. For SymCpAbe, the data rate in which the *Gateway* is overloaded is higher (64 data/second). Specifically, with the most SymCpAbe lightweight case (a 1-attribute POL), the relative percentage of published events is 39% compared to 5% in the case of the CP-ABE approach. Even with a 10-attributes POL, our solution achieves a relative percentage of 24%, against the 5% with CP-ABE. Consequently, from these results, it is demonstrated SymCpAbe scheme allows to protect and publish a greater amount of events in contrast with the CP-ABE approach. This is specially relevant in IoT-enabled scenarios, such as the considered use case, in which large amounts of data need to be protected.

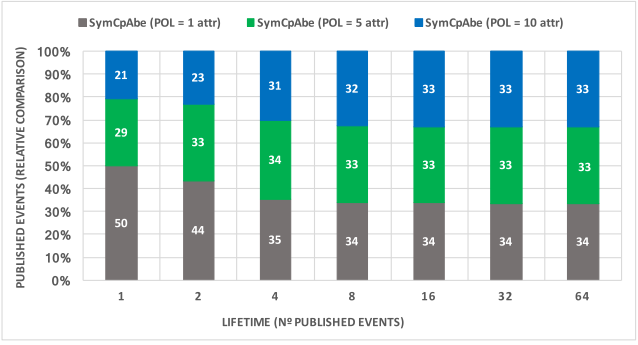

FIGURE 7. Percentage of published events by the *Gateway*, varying the $SYMK_{lifetime}$.

Furthermore, Figure 7 shows the relative comparison of the number of published events by the *Gateway* (Phase 3) according to the $SYMK_{lifetime}$, with the purpose to demonstrate how this parameter affects our approach. In this sense, when the $SYMK_{lifetime}$ is decreased, the relative percentage of published events is mainly affected by the number of attributes in POL (50% using policies including 1 attribute compared to 21% with policies including 10 attributes). As the $SYMK_{lifetime}$

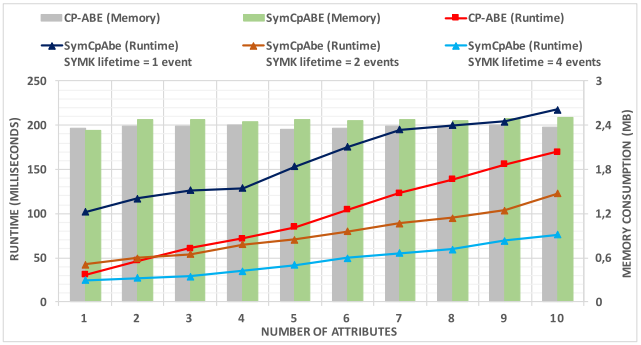

FIGURE 8. Memory and runtime for CP-ABE and SymCpAbe by the *Service*.

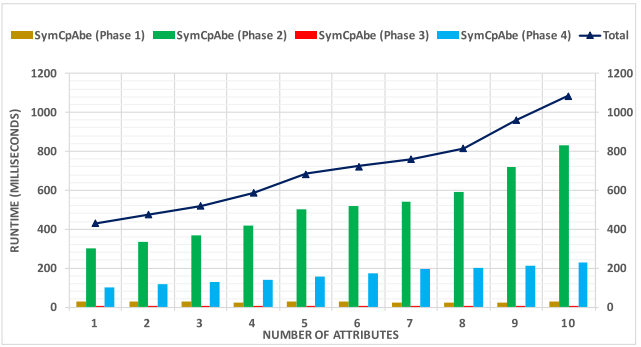

FIGURE 9. Runtime for the SymCpAbe approach.

is increased, this percentage becomes constant regardless of the number of attributes in POL. This is due to the reduction of CP-ABE encryption operations when the $SYMK_{lifetime}$ increases, so that the *Gateway* is able to encrypt and publish more incoming data.

B. DATA EVENT RETRIEVAL PERFORMANCE

This phase comprises the operations required by *Services* to get the data from the *Publish/Subscribe Broker* and decrypt them. In particular, it covers Phase 4 in the case of SymCpAbe, while for CP-ABE, the *Services* are responsible for decrypting each data by using such scheme. Thus, Figure 8 shows the required memory consumption and runtime by considering CP-ABE and SymCpAbe approaches according to the number of attributes in POL. While in the case of CP-ABE the memory consumption remains constant, in the case of SymCpAbe, this value is slightly increased. Indeed, with the direct application of CP-ABE, the *Service* only needs to perform the CP-ABE decryption operation to get access the data of event. In contrast, using our approach, it should firstly contact the *Symmetric Key Database* to get the SYMK that was used to encrypt such data. Moreover, we have considered different cases according to the value of $SYMK_{lifetime}$ for the runtime required by *Services* to retrieve the data. As shown, only for the *"worst case"* for SymCpAbe, the performance of the CP-ABE approach is better, since for that case, the *Service* should get a new SYMK for each received event. Indeed, when the $SYMK_{lifetime}$ is increased, the performance of SymCpAbe is better than CP-ABE. This is because the most expensive operations (i.e. getting the SYMK and decrypting it by using CP-ABE) are only required in case that a new SYMK is used by the *Gateway* to protect the data.

Finally, Figure 9 shows the runtime required for the different phases of SymCpAbe *"worst case"* by varying the number of attributes in POL. Taking into account the results obtained in Figure 5, the runtime for the whole SymCpAbe approach (i.e. also including runtime spent by the *ABES* and the *Symmetric Key Database* to carry out their functionality) is up to 6 times lower than the direct application of CP-ABE from the *Gateway* side (for a 10-attribute POL, 1086 ms and 6525 ms, respectively). Hence, it is demonstrated that our scheme represents an efficient and scalable approach to be used on scenarios where large data amounts need to be protected. In next section, certain security considerations are discussed regarding the suitability of our proposed approach.

VII. SECURITY ANALYSIS

As already mentioned, SymCpAbe represents a scalable encryption approach, which is intended to achieve a trade-off between performance and the fulfillment of different security properties. It should be noted that the inclusion of additional components compared with the direct application of CP-ABE gives rise to further security aspects to be considered. Based on other related works ([26], [40], [41]), we have analysed the following security properties over our proposed approach:

Key Escrow [42]: By using SymCpAbe, the *ABES* has the keys that are used by gateways to encrypt data. Consequently, it could access the data from the *Publish/Subscribe Broker* in case they are required. This fact represents an inherent aspect to be considered for the adoption of ABE-based schemes [43] since the end users' privacy could be threatened. In this sense, we have considered the *ABES* as a semi-trusted service (i.e. honest but curious), so it does not confabulate with other entities to use such data with malicious intent. Additionally, this service could still be authorized by the *Symmetric Key Database* and the *Publish/Subscribe Broker* to access both the encrypted keys and the encrypted data. While this point has not been addressed by our approach, authorization aspects to publish or obtain keys and data represent part of our future work. Thus, authorization models based on our access control approach based on capabilities [44] could be further integrated. In addition, in order to overcome the problems arising from the use of a single entity for the CP-ABE encryption (i.e. the *ABES*), alternative approaches based on outsourcing CP-ABE operations could be applied, such as [45] and [46]. Specifically, in our case, the *Gateway* would be able to outsource the CP-ABE encryption of the SYMK without the need to disclose such key itself. However, note that even in this situation, if cryptographic operations are outsourced to more powerful entities, network overhead could still involve a significant issue in certain scenarios.

Security Level [47]: SymCpAbe is independent of the length of the cryptographic keys that are to be employed.

While we have considered the P-256 curve (i.e. a 128-bit security level) for the ECDHE algorithm, other curves providing more security level can be used (e.g. P-384 or P-521). Moreover, it should be pointed out that the SYMK encryption via CP-ABE uses type A pairings, which are built on the supersingular curve y2 = x3 + x over the field Fp for some prime p = 3mod4. In this case, let p be the prime order of Fp, and E(Fp), the additive group of points of affine coordinates (x, y) with x, y in Fp that satisfy the curve equation, q represents the order of the cyclic subgroup of interest in E(Fp). Under these considerations, evaluation results has been carried out with |p| = 512, |q| = 160, obtaining a security level of 80-bit. The selection of these values for q and p is based on the results shown in [20], although other values for CP-ABE cryptographic operations could be considered to increase that level (e.g. 112-bit or 128-bit) [22]. Note that, in case of using higher security levels, the performance of SymCpAbe would be further improved regarding the direct application of CP-ABE.

Perfect Forward Secrecy [48]: Regarding this aspect, the SymCpAbe approach establishes SYMKs by using the ephemeral version of the ECDH algorithm. In addition, we assume that these keys are removed from *Gateways*, the *ABES*, the *Symmetric Key Database* and *Services* once they are not valid. This way, this property is assured since, even if a SYMK is compromised, all data encrypted with previous SYMKs will not be accessible. In this sense, each SYMK has an associated lifetime ($SYMK_{lifetime}$), which is to be established according to different practical aspects depending on the scenario. On the one hand, it should be long enough, so the *Gateway* does not need to frequently generate new SYMKs to protect data. On the other hand, it is required to be short enough so that an attacker is not encouraged to perform brute-force or dictionary attacks in order to infer the SYMK being employed.

Collusion Resistance [49]: In SymCpAbe, the resistance against collusion attacks is inherited from the CP-ABE scheme [9]. This property guarantees that two attackers cannot combine their SKs to compute a new CP-ABE private key representing the union of their attributes in order to decrypt a SYMK, and consequently, retrieving the encrypted data.

Data Access Control: Under the proposed SymCpAbe approach, users are enabled to define how their information is to be shared and under which circumstances. This property is inherited from the CP-ABE scheme that allows to define the combination of identity attributes that must be satisfied by intended receivers. In our approach, SYMKs are CP-ABE encrypted under a specific POL, so AES encrypted data will be accessed whenever the attribute set of a *Service* satisfies such POL. This way, users maintain the access control over their information, assuring it will be only recovered by authorized entities.

As described above, in addition to providing an efficient and scalable approach for data protection, SymCpAbe addresses major security aspects that must be considered in IoT-enabled scenarios. Furthermore, unlike other recent proposals [24], [26], SymCpAbe makes use of the JWA and JWK proposals from the IETF, in order to represent the exchanged cryptographic material, thereby improving the interoperability of our scheme. Additionally, while this cryptographic material is protected by using TLS, other emerging alternatives, such as the *JSON Object Signing and Encryption* (JOSE) [50], could be integrated to the SymCpAbe scheme to come up with a more comprehensive and effective approach in order to cope with the security challenges associated to data protection in IoT-enabled scenarios.

VIII. CONCLUSIONS AND FUTURE WORK

In recent years, with the emergence of IoT, smart buildings are been established as the evolution of residential and industrial buildings already existing. In these nascent environments, produced data can be obtained, communicated, and processed to allow services to make decisions accordingly. While these data-driven applications can have a significant impact on the daily activity of a smart city, such impact may have an undesirable effect in terms of users' privacy, if appropriate security mechanisms are not implemented. To mitigate the challenges associated with the protection of large amounts of data in such environments, this work has presented a novel scheme (SymCpAbe) that combines the advantages of the symmetric and attribute-based encryption schemes. SymCpAbe has been compared to a pure CP-ABE approach adopted in other current proposals, by deploying both scheme on a real smart building scenario in order to evaluate the performance of our proposal. Thus, evaluation results demonstrate SymCpAbe provides a more efficient and flexible solution to ensure the protection of sensitive data while scalability is preserved. Future work focuses on designing and developing a new mechanism to distribute CP-ABE encryption and decryption in different edge nodes to cooperatively perform these resource-demanding cryptographic operations. Additionally, we plan to analyse and deploy such mechanism on different IoT-enabled scenarios, including Industry 4.0 use cases.

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
[2] EuroACE. *Smart Buildings: Energy Efficiency First*. Accessed: Jul. 31, 2017. [Online]. Available: http://euroace.org/wp-content/uploads/2015/10/EA_Smart_Buildings_Feb_2017_Final.pdf
[3] *Bringing Big Data to the Enterprise*. Accessed: Jul. 31, 2017. [Online]. Available: https://www-01.ibm.com/software/sg/data/bigdata
[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[5] F. Zafari, I. Papapanagiotou, and K. Christidis, "Microlocation for Internet-of-Things-equipped smart buildings," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 96–112, Feb. 2016.
[6] T. Weng and Y. Agarwal, "From buildings to smart buildings—Sensing and actuation to improve energy efficiency," *IEEE Design Test Comput.*, vol. 29, no. 4, pp. 36–44, Aug. 2012.
[7] S. Pérez, J. L. Hernández-Ramos, A. F. Skarmeta, D. Pedone, D. Rotondi, and L. Straniero, "A digital envelope approach using attribute-based encryption for secure data exchange in IoT scenarios," in *Proc. IEEE 1st Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.

[8] A. Kahate, *Cryptography and Network Security*. New York, NY, USA: McGraw-Hill, 2013.
[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.
[10] M. Jones, *JSON Web Key (JWK)*, document RFC 7517, May 2015.
[11] *JSON Web Algorithms (JWA)*, document RFC 7518, May 2015.
[12] P. S. Makeshwar and G. Borse, "Improving security in group based data sharing using multicast key agreement," *Int. J. Eng. Sci.*, vol. 7, no. 2, 2017, Art. no. 4468.
[13] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 309–329, 2003.
[14] Z. Zhou and D. Huang, "An optimal key distribution scheme for secure multicast group communication," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.
[15] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.
[16] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Secure and scalable cloud-based architecture for e-health wireless sensor networks," in *Proc. IEEE 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2012, pp. 1–7.
[17] P. Picazo-Sanchez, J. E. Tapiador, P. Peris-Lopez, and G. Suarez-Tangil, "Secure publish-subscribe protocols for heterogeneous medical wireless body area networks," *Sensors*, vol. 14, no. 12, pp. 22619–22642, 2014.
[18] M. Qiu, K. Gai, B. Thuraisingham, L. Tao, and H. Zhao, "Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry," *Future Generat. Comput. Syst.*, vol. 80, pp. 421–429, Mar. 2016.
[19] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 135–146, 2009.
[20] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 725–730.
[21] N. Koblitz and A. Menezes, "Pairing-based cryptography at high security levels," in *Proc. IMA Int. Conf. Cryptograp. Coding.*, 2005, pp. 13–36.
[22] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," in *Proc. Workshop IoT Challenges Mobile Ind. Syst. ACM*, 2015, pp. 49–54.
[23] L. Touati, Y. Challal, and A. Bouabdallah, "C-CP-ABE: Cooperative ciphertext policy attribute-based encryption for the Internet of Things," in *Proc. Int. Conf. Adv. Netw. Distrib. Syst. Appl. (INDS)*, 2014, pp. 64–69.
[24] J. Xu, Q. Wen, W. Li, and Z. Jin, "Circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 119–129, Jan. 2016.
[25] M. Morales-Sandoval and A. Diaz-Perez, "DET-ABE: A java API for data confidentiality and fine-grained access control from attribute based encryption," in *Proc. IFIP Int. Conf. Inf. Secur. Theory Pract.*, 2015, pp. 104–119.
[26] D. Thatmann, S. Zickau, A. Förster, and A. Küpper, "Applying attribute-based encryption on publish subscribe messaging patterns for the Internet of Things," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst. (DSDIS)*, Dec. 2015, pp. 556–563.
[27] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.
[28] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Elsevier Future Generat. Comput. Syst.*, vol. 49, pp. 104–112, Aug. 2015.
[29] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," in *Proc. IEEE 5th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Apr. 2015, pp. 746–751.
[30] J. M. Bohli, A. Skarmeta, M. V. Moreno, D. A. Garc, and P. Langendorfer, "Smartie project: Secure IoT data management for smart cities," in *Proc. Int. Conf. Recent Adv. Internet Things (RIoT)*, 2015, pp. 1–6.
[31] *Oasis Advanced Message Queuing Protocol (AMQP) Version 1.0*, OASIS Standard, 2012. [Online]. Available: http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf
[32] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (COAP)*, document RFC 7252, Jun. 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7252.txt
[33] A. Banks and R. Gupta, *MQTT Version 3.1.1*, OASIS standard, 2014.
[34] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, document RFC 5246, Aug. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5246.txt
[35] E. Rescorla and N. Modadugu, *Datagram Transport Layer Security Version 1.2*, document RFC 6347, Jan. 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6347.txt
[36] D. McGrew, K. Igoe, and M. Salter, *Fundamental Elliptic Curve Cryptography Algorithms*, document RFC 6090, Feb. 2011.
[37] E. Barker, L. Chen, A. Roginsky, and M. Smid, "Recommendation for pairwise key establishment schemes using discrete logarithm cryptography," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST Special Publication 800-56, 2006.
[38] G. Klyne and C. Newman, *Date and time on the Internet: Timestamps*, document RFC 3339, 2002. [Online]. Available: https://tools.ietf.org/html/rfc3339
[39] J. Wang. (2012). *Ciphertext-Policy Attribute-Based Encryption Library*. Accessed: Jul. 31, 2017. [Online]. Available: https://github.com/junwei-wang/cpabe
[40] G. Lin, H. Hong, and Z. Sun, "A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing," *IEEE Access*, vol. 5, pp. 9464–9475, 2017.
[41] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: Enabling security and patient-centric access control for e-health in cloud computing," *Int. J. Secur. Netw.*, vol. 6, nos. 2–3, pp. 67–76, 2011.
[42] H. Abelson *et al.*, "The risks of key recovery, key escrow, and trusted third-party encryption," *World Wide Web J.*, vol. 2, no. 3, pp. 241–257, 1997.
[43] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.
[44] J. L. Hernández-Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid, "Toward a lightweight authentication and authorization framework for smart objects," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 690–702, Apr. 2015.
[45] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014.
[46] M. Green, S. Hohenberger, and B. Watersetal, "Outsourcing the decryption of ABE ciphertexts," in *Proc. USENIX Secur. Symp.*, 2011, p. 34.
[47] E. B. Barker, W. C. Barker, W. E. Burr, W. T. Polk, and M. E. Smid. (2012). *NIST Publication*. [Online]. Available: https://www.nist.gov/publications/recommendation-key-management-part-1-general-revision-3
[48] H. Krawczyk, "Perfect forward secrecy," in *Encyclopedia Cryptography Security*. New York, NY, USA: Springer, 2005, pp. 457–458.
[49] Z. Zhu and G. Cao, "Toward privacy preserving and collusion resistance in a location proof updating system," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 51–64, Jan. 2013.
[50] M. Miller, *Examples of Protecting Content Using JSON Object Signing and Encryption (JOSE)*, document RFC 7520, May 2015.
[51] J. Bethencourt, A. Sahai, and B. Waters. (2011). *CPABE Toolkit*. [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

SALVADOR PÉREZ received the B.Sc. degree in computer science and M.Sc. degree in new technologies in computer science from the University of Murcia, Spain, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information and Communications Engineering. He is currently a Researcher with the Department of Information and Communications Engineering, University of Murcia. His main research interests include defining scalable approaches to manage security and privacy on groups of IoT devices and services.

JOSÉ L. HERNÁNDEZ-RAMOS received the M.Sc. and Ph.D. degrees in computer science from the University of Murcia, Spain. Since 2013, he has been a Research Fellow with the Department of Information and Communications Engineering, University of Murcia, where he has participated in different european research projects, such as SocIoTal and SMARTIE. His research interests are mainly related to the application of security and privacy mechanisms for the Internet of Things.

SARA N. MATHEU-GARCÍA received the B.S. degree in mathematics and B.S. and M.S. degrees in computer science from the University of Murcia, in 2010 and 2011, respectively, where she is currently pursuing the Ph.D. degree. Since 2016, she has been a Pre-Doctoral Researcher with the Department of Information and Communications Engineering, University of Murcia, where she is collaborating in the European research project ARMOUR. Her main research interests are related to the security certification for the Internet of Things.

DOMENICO ROTONDI received the degree (*cum laude*) in physics. In 2005, he joined TXT e-Solutions SpA, where he supports the Supply Chain Management Division on aspects related to applications security and reliability analysis (threats analysis) and TXT products certification. In the TXT Innovation and Research Division, he was involved in many research activities focused on Identity Management, authentication and security issues, and also new technological approaches and technologies (SOA, semantic web service, Internet of Things and cloud computing, business process management standards). Since 2014, he has been cooperating with the FINCONS Group International Business Development and Innovation Unit continuing its research and development activities in the same fields. He has been a Project and Technical Manager in several research projects, such as FP6 MyTreasury, FP7 SHIELDS, FP7 GEMOM, FP7 COIN, FP7 NMP TIPSS and CORENET, FP7 IoT@Work, FP7 MUSES, FI-PPP FITMAN, H2020 FoF09 BEinCPPS, H2020 FoF05 PSYMBIOSYS, and H2020 CULT-COOP-11 CITADEL. He was with the Industrial Advisor Board of the NESSOS NoE and was a FITMAN Representative at the FI-PPP Architectural Board. He has a Postgraduate Scholarship in computer networks, distributed systems, and telematics services.

ANTONIO F. SKARMETA received the the B.S. (Hons.) degree in computer science from the University of Murcia, Spain, the M.S. degree in computer science from the University of Granada, and the Ph.D. degree in computer science from the University of Murcia, Spain. Since 2009, he has been a Full Professor with the Department of Computer Science, University of Murcia. He has published over 200 international papers. His main research interests include integration of security services, identity, IoT, and smart cities. He has been a member of several program committees.

LEONARDO STRANIERO received the bachelor's degree in computer science from the University of Bari, Bari, Italy, in 2008. He has collaborated on several projects in the fields of health, energy, insurance, and public administration. He has also several years of experience on research and development teams by taking part in national or European projects in the manufacturing and health fields. His involvement in these research and development projects was essentially focused on security issues and development of distributed interoperable services. He joined as a Senior Programmer at FINCONS SpA in 2015, where he collaborates with the International Business Development and Innovation BU.

DIEGO PEDONE received the bachelor's degree in computer science and digital communication from the University of Bari, Bari, Italy, in 2013, with a thesis done within an internship at the I&T Group s.r.l, Bari. In 2013, he joined FINCONS SpA as a Developer for manufacturing and public administration customers. Since 2015, he has been collaborating with the FINCONS International Business Development and Innovation BU in national and European projects.