Integration of Attribute-Based Encryption and IoT: An IoT Security Architecture

Author:

Ziyad Elbanna

Department of Computer and Systems Sciences

Degree project 30 HE credits
Computer and Systems Sciences
Degree project at the master level
Spring term 2023
Supervisor: Yuhong Li

Stockholm University

Abstract

Services relying on internet of things (IoTs) are increasing day by day. IoT makes use of internet services like network connectivity and computing capability to transform everyday objects into smart things that can interact with users, and the environment to achieve a purpose they are designed for. IoT nodes are memory, and energy constrained devices that acquire information from the surrounding environment, those nodes cannot handle complex data processing and heavy security tasks alone, thus, in most cases a framework is required for processing, storing, and securing data. The framework can be cloud-based, a publish/subscribe broker, or edge computing based.

As services relying on IoT are increasing enormously nowadays, data security and privacy are becoming concerns. Security concerns arise from the fact that most IoT data are stored unencrypted on untrusted third-party clouds, which results in many issues like data theft, data manipulation, and unauthorized disclosure. While some of the solutions provide frameworks that store data in encrypted forms, coarse-grained encryption provides less specific access policies to the users accessing data. A more secure control method applies fine-grained access control, and is known as attribute-based encryption (ABE). This research aims to enhance the privacy and the security of the data stored in an IoT middleware named network smart objects (NOS) and extend its functionality by proposing a new IoT security architecture using an efficient ABE scheme known as key-policy attribute-based encryption (KP-ABE) along with an efficient key revocation mechanism based on proxy re-encryption (PRE).

Design science research (DSR) was used to facilitate the solution. To establish the knowledge base, a previous case study was reviewed to explicate the problem and the requirements to the artefact were elicited from research documents. The artefact was designed and then demonstrated in a practical experiment by means of Ubuntu operating system (OS). Finally, the artefact’s requirements were evaluated by applying a computer simulation on the Ubuntu OS.

The result of the research is a model artefact of an IoT security architecture which is based on ABE. The model prescribes the components and the architectural structure of the IoT system. The IoT system consists of four entities: data producers, data consumers, NOS, and the TA. The model prescribes the new components needed to implement KP-ABE and PRE modules. First, data is transferred from data producers to NOS through secure hypertext transfer protocol (HTTPS), then the data is periodically processed and analyzed to obtain a uniform representation and add useful metadata regarding security, privacy, and data-quality. After that, the data is encrypted by KP-ABE using users’ attributes. PRE takes place when a decryption key is compromised, then the ciphertext is re-encrypted to prevent it’s disclosure. The evaluation results show that the proposed model improved the data retrieval time of the previous middleware by 32% and the re-encryption time by 87%. Finally, the author discusses the limitations of the proposed model and highlights directions for future research.

Keywords*: Internet of things, Attribute based encryption, Fine-grained access control, Key revocation, Proxy re-encryption.*

Synopsis

Background

The topic of the thesis belongs to the cybersecurity area within computer and systems sciences. The thesis proposes a model which prescribes the components and functionalities required to introduce KP-ABE and PRE in NOS middleware to manage access to IoT data in IoT applications and demonstrates the artefact on a smart home scenario.

IoT refers to the interconnection of things, devices, and networks to share data between users, and achieve a purpose the object is designed for. Smart houses are an application domain in the field of IoT where it consists of home controllers and sensors that generate data, such as electricity, remote monitoring of temperature, humidity, and video streaming.

Problem

IoT data are prone to security and privacy issues like data theft, data manipulation, and unauthorized disclosure. Although many frameworks employ encryption techniques applying coarse-grained access control, this access control method is not sufficient as it does not enable high level of specificity to whom can access the data. Another encryption method is more promising in maintaining access only to authorized users using a higher level of specificity for data access rules, this method is known as ABE.

IoT smart homes need to use an IoT system that encrypts data using ABE methods that does not result in a significant data retrieval delay. Moreover, to enhance security, IoT frameworks shall use an efficient method for key revocation like PRE to prevent data theft, and data manipulation in case the key is compromised, or a user is revoked.

Research Question

How can we enhance the security and efficiency of ABE in IoT applications?

Method

The author applied DSR strategy for conducting the thesis research. Firstly, the problem was explicated by reviewing a case study that identifies the issues of an IoT middleware, and by reviewing documents of similar IoT issues. Second, the requirements were elicited by reviewing documents of IoT frameworks and efficient re-encryption methods, privacy enhancement, and efficient ABE schemes. Third, the artefact was designed and developed using the knowledge acquired from the documents and is presented as an initial design. Fourth, the author demonstrated the artefact's constructs on a laptop installing two Ubuntu operating systems using datasets of IoT smart homes. Fifth, evaluation was done by applying an ex-ante approach with artificial evaluation strategy on a computer simulation.

Result

In design science, research artefacts include models, methods, instantiations, and constructs. The proposed artefact is a model that prescribes the functionalities and components of the IoT system that is based on ABE and PRE. The evaluation results show that the new model improves the data retrieval time of the previous model by 32% and the re-encryption time by 87% compared to the old model.

Discussion

The thesis fills the gaps identified in previous architectures and schemes used in IoT scenarios which are reviewed in the thesis. An IoT Security model was constructed which uses KP-ABE module which are more time-efficient than the previously used technique (i.e., CP-ABE). Moreover, the model uses PRE method as a method for key revocation in case the key is compromised, or a user is revoked.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IoT – Internet of Things
NOS – Networked Smart Objects
OS – Operating system
ABE – Attribute Based Encryption
ICT – Information and Communications Technology
IS – Information Systems
QOS – Quality of Service
CP-ABE – Ciphertext policy Attribute Based Encryption
KP-ABE – Key policy Attribute Based Encryption
ECC – Elliptic Curve Cryptography
DSR – Design Science Research
IMRAD – Introduction, methods, results, and discussion
TA – Trusted Authority
RFID – Radio Frequency Identification
S-IoT – Social Internet of Things
HTTP – Hypertext Transfer Protocol
HTTPS – Secure Hypertext Transfer Protocol
PRE – Proxy Re-encryption
KGC – Key Generation Center
MQTT – Message Queue Telemetry Transport
SMQTT – Secure Message Queue Telemetry Transport
KPI – Key Performance Indicator

1 Introduction

This chapter offers an introduction to the thesis and introduces the problem, the research question, and delimitations.

1.1 Background

The research problem in this thesis appears in the field of data transmission and storage security in IoT applications. Data storage privacy and security belongs to the cybersecurity area of computer engineering. The thesis aims to reduce the privacy and security issues in data sharing in IoT applications by proposing a new IoT security architecture based on a cryptographic method known as attribute-based encryption (ABE). Moreover, the thesis solves some limitations found in a previous data sharing framework which employs ABE methods. An IoT data sharing framework is a system which deals with data sent from heterogenous data sources by processing and collecting data in an automated manner (Sicari et al., 2016). This thesis contributes to the multitude of information regarding ABE of data stored in the IoT middleware known as: Networked smart objects (NOS) which was previously designed by Rizzardi et. al., (2016). Furthermore, the thesis's proposed model adds new functionalities to the previously designed middleware's structure. The model can be instantiated and used as an alternative in many IoT applications.

In the thesis, the author demonstrated the model's functionalities using the same datasets used by Sicari et al. (2020) and evaluated its performance. According to Sicari, the middleware employs ciphertext-policy attribute-based method (CP-ABE) which provides no efficient re-encryption method during key revocation, and takes a lot of time for encryption, decryption, and key-generation (2021). This project's main activities include enhancing the time efficiency of the ABE operations in the model proposed by Sicari by using a KP-ABE scheme which is more efficient than the previously used ABE scheme. Another activity is to enhance the data security of encrypted data by adding an efficient re-encryption mechanism based on PRE to prevent data theft, data manipulation, and unauthorized access to ciphertexts in case the key is compromised, or a user is revoked. In the smart home scenario, the stakeholders are data producers, the data storage entity, data consumers, and the TA. Data producers in the IoT application are house sensors and controllers, while data consumers are users who consume data in the application, such as the tenant of the house, the landlord of the house, and the guest of the house. The trusted authority is the entity which creates, generates, and distributes the secret keys and public parameters. The data storage entity is the NOS middleware.

1.1.1 Internet of Things overview

The concept of IoT was first proposed by Kevin Ashton in 1999. Kevin referred to IoT as uniquely identifiable interoperable connected objects identified with the help of radio-frequency identification (RFID) technology that uses tags that are inserted into equipment and contain information that identifies that equipment (Kabachinski, 2005). However, there is no single universal definition for IoT. IoT was generally defined as "scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention" (Rose, 2015, p.1).

Objects using internet connectivity can range from vehicles, consumer products, industrial components, and other everyday objects that will change the way we work, live and play. Other technologies may also refer to IoT like sensors technologies using RFID, smart things, nanotechnology, and miniaturization (Mukhopadhyay, 2014). Moreover, some researchers like Li et al. (2014) refer to IoT as "inter-connected world-wide network based on sensory, communication, networking, and information processing technologies" (p. 244), which might be the new version of information and communications technology (ICT). Combining computers, sensors, and networks to control and monitor devices is not a new concept as it has been used for decades. However, the integration of several technology market trends nowadays brought IoT closer to widespread reality and made it a very important topic, these trends include ubiquitous connectivity, the rise of cloud computing, and advances in data analytics (Rose, 2015).

1.1.2 Internet of Things applications

Nowadays, IoT is expected to be incorporated into products, services, and operations in many industrial sectors including automotive, healthcare, manufacturing, and consumer electronics (Rose, 2015). In their report, the McKinsey global institute describes the potential applications where IoT is expected to create value for industry and users. The potential applications are described in terms of "settings" in Table 1-1.

"Settings" for IoT Applications		
Setting	**Description**	**Example**
Human	Devices attached or inside the human body	Devices (wearables) to monitor and maintain human health and wellness
Home	Buildings where people live	Home controllers and security systems
Retail Environments	Spaces where consumers engage in commerce	Stores, banks, restaurants, arenas
Offices	Spaces where knowledge workers work	Office buildings
Factories	Standardized production environments	Places with repetitive work routines, including hospitals and farms; operating efficiencies, optimizing equipment use and inventory
Worksites	Custom production environments	Mining, oil and gas, construction, operating efficiencies, predictive maintenance, health, and safety
Vehicles	Systems inside moving vehicles	Vehicles including cars, trucks, ships, aircraft, and trains; condition-based maintenance, usage-based design, pre-sales analytics
Cities	Urban environments	Public spaces and infrastructure in urban settings; adaptive traffic control, smart meters, environmental monitoring, resource management
Outside	Between urban environments (and outside other settings)	Outside uses include railroad tracks, autonomous vehicles (outside urban locations), and flight navigation; real-time routing, connected navigation, and shipment tracking

Table 1-1: "Settings" for IoT applications from Manyika et al. (2015)

IoT technologies are key enablers for a multitude of applications in diverse fields like smart homes, smart cities, digital health, industrial automation, and supply-chain (Rasori et al., 2022). Some researchers like Li et al. (2014) list IoT applications based on the field they are used in. Li lists IoT applications as: industrial applications, social applications, healthcare applications, infrastructure, and security and surveillance.

Industrial Applications

IoT can be used to manage fine-grained applications like online payment, aggregated quality of service (QoS), and critical data storage. IoT is able to improve the efficiency of business transactions and provide more efficient services in the field of industry (Li et al., 2014).

Social IoT

Atzori proposed an idea where IoT devices can be connected to social networks (Atzori et al., 2011). The proposition is to create a world where everything is connected through the internet. Social IoT (SIoT) has attracted the attention of many scientists and researchers around the world due to the enormous number of social network users. Hernandez-Castro et al. suggested the integration of IoT with existing social networks such as Facebook and Twitter (2013).

Healthcare

According to Boyi, the healthcare field is a very important area for applying IoT (Boyi et al., 2014). IoT devices such as medical sensors are used to monitor some medical parameters such as body temperature and blood pressure. The values of the parameters are then transferred to the electronic health record to be stored, which is usually a cloud-based application used to manage, process, and store health record information of patients to be used by several healthcare institutions and professionals (Edemacu et al., 2019).

Infrastructure

IoT is used in many infrastructure areas such as smart homes, environmental monitoring, and smart cities. It can be used to improve the quality of buildings and reduce waste.

Security and Surveillance

IoT components and technologies can be used as an extra layer of protection and security to devices, as the security techniques applied in the conventional networks between IoT devices are not sufficient (Kang et al. 2014). “Security technologies should provide strong protection for all levels of system components at all stages: from sense layer to interface layers, from identification to service provision, and from RFID tags to IT infrastructure” (Li et al. 2014, p. 255). As a result, devices connecting to the internet are in constant increase day after day, attackers are finding new security vulnerabilities day by day and exploit them. This raises a lot of security and privacy issues which are discussed in section 1.1.3.

1.1.3 Security and privacy considerations of IoT

Security Issues

As users of the internet, we need to have a high degree of trust that the internet, its applications, and the devices connected to it are secure enough to do the kind of activities we want to do online. As we increasingly connect devices to the internet, opportunities to exploit security vulnerabilities grow. For example, poorly secured IoT devices could serve as an entry point for cyber-attacks by allowing attackers to change their program functionality or cause them to malfunction. Malfunctioning can lead to more security vulnerabilities. IoT devices' competitive costs and technical constraints challenge their manufacturers to adequately design security features, creating security and long-term maintainability vulnerabilities greater than their traditional computer counterparts (Rose, 2015). Some devices such as a smart TV, or a smart refrigerator can be unplugged if they get compromised in a cyberattack, but a smart utility power meter or a traffic control system cannot be turned off if it is compromised in a cyber-attack.

This is why security of IoT services and devices should be considered a critical issue as we rely on these devices for essential services, and their behavior may have global impact. The security challenges of IoT devices include the following:

- IoT devices such as sensors and consumer items, are designed to be deployed at a massive scale, thus, the quantity of interconnections between these devices is very large. Therefore, existing tools, methods, and strategies associated with IoT security will need new considerations.
- Many IoT devices are intentionally designed without any ability to be upgraded, or the upgrade process is cumbersome or impractical. Therefore, devices are left vulnerable to new cybersecurity threats.
- Most IoT devices operate in a manner where the user cannot oversee the internal workings of the device or the data streams they produce. Therefore, resulting in a security vulnerability when a user thinks it is working in some way but, it is working in another way like collecting more data than needed.
- Some IoT devices are kept in a place where physical security is impossible to achieve. Thus, the attacker may have direct access to the devices. New design innovations ensuring security will need to be considered.

Privacy Issues

Respect for privacy expectations and rights is essential to ensure trust on the internet, it also impacts the ability of individuals to communicate, and connect in meaningful ways. These expectations and rights are sometimes framed in terms of ethical data handling, which emphasizes the importance of respecting people's expectation of privacy and the fair use of their data (Robin, 2014). IoT can challenge these rights of privacy.

Characteristics of IoT devices re-define the debate about how data can be collected, processed, analyzed, and protected. For example:

- IoT devices have no user interface to configure privacy preferences. Thus, in many IoT applications users have no control or knowledge about how their data will be handled.

- Some IoT devices collect information about users with a very high degree of specificity; correlation of these data can create detailed profiles of individuals which can create potential for discrimination and other harms.

IoT creates unique challenges to user privacy. Thus, new strategies need to be developed to respect individual privacy.

1.1.4 ABE Challenges in IoT

In IoT, data exchange occurs between two main entities: (i) Data producers (nodes, and sensors), and (ii) Data consumers (users). Data is mostly transferred, stored, and encrypted with the help of IoT data sharing frameworks. IoT frameworks manage and process data from heterogenous data sources, frameworks store and transfer data in an automated way (Sicari et al., 2016). Frameworks can be cloud-based, publish/subscribe-based, or edge-computing based. The most crucial challenge in constructing an IoT system lies in the lack of common and standardized software framework that can be used in all the applications (Rizzardi et al., 2015).

Introducing a secure data sharing framework for IoT requires validating data sources, thus ensuring data accuracy, and validity. Data quality and integrity are other factors that should also be analyzed and processed before the data can be accessed by the consumers (Rizzardi et al., 2015). Moreover, for ABE, additional KPIs should be evaluated like execution time, CPU load, and memory consumption (Wang et al., 2014; Selar et al., 2017). Maintaining the required features for enhancing data security and ABE efficiency and merging all of them in one framework is a challenging issue and is ongoing a lot of research effort (Singh et al., 2015; Sanchez et al., 2014; Yao et al. 2014; Sicari et al., 2020).

The IoT middleware proposed by Sicari et al. (2020) employed CP-ABE for data encryption. The middleware combined privacy, security, and data-quality by using analysis modules, as well as fine-grained encryption by using ABE. However, the middleware's performance evaluation of data retrieval time, and memory consumption showed that it did not suit the needs of IoT users and that it can be improved; due to the time delay it caused and the research indicated that the efficiency of re-encryption during key revocation needs to be improved (Sicari et al., 2020).

In healthcare applications, lightweight is an essential feature in the used IoT devices (Sanchez et al., 2014). To obtain the lightweight feature of the scheme, the authors constructed a framework that implements a new algorithm that uses only-AND access gates. The problem with this scheme is that it restricted the types of policies used in different IoT applications and services and limited the access control flexibility.

Another framework proposed by Singh et al.'s model (2015) employed ABE techniques on MQTT broker over MQTT protocol and employed an MQTT broker as a trusted key authority as well as a communication enabling entity that stores messages. However, the author maintained the lightweight property of IoT devices by not implementing a revocation method. This limits the ability of the framework if a user is revoked, or a key is compromised. A thorough review of the existing ABE frameworks and schemes is documented in section 2.2.

1.2 Research Problem

The main problem developed from the fact that potential security vulnerabilities started to appear because the interconnections of IoT, and networks are now deployed in many business sectors like infrastructure, security, surveillance, healthcare, home, and industrial applications. Poorly secured IoT devices could serve as an entry point for an attacker to re-program a device or cause malfunction or exploit vulnerabilities to access IoT data and steal it (Rose, 2015). Although, there are encryption techniques based on coarse-grained access control developed to secure IoT data, they don't provide strict rules to whom can access these data, thus, other encryption techniques based on fine-grained access control like ABE are more promising in providing more strict access control rules. Unauthorized access to data may result in data theft, data manipulation, and data disclosure. The effects of these issues may be detrimental to the entities that employ IoT, as users will stop using IoT devices due to privacy breaches.

The precise problem was extracted from a case study where the ABE encryption schemes chosen for the IoT framework were shown to be inefficient and lacking some functionalities. In the smart home scenario examined by Sicari et al. (2020), a middleware employing ABE was constructed, which implements CP-ABE. The performance evaluation of CP-ABE in the IoT scenario has shown a noticeable delay, moreover, the scheme implemented by Sicari did not allow for efficient re-encryption during key revocation. IoT home scenarios often use IoT devices which are resource-constrained, battery-powered and have limited connectivity and perform lightweight operations. Thus, they require encryption and data storage schemes that consume less energy, introduce less data overhead, and offer a quicker response to IoT users. In these cases, both KP-ABE and CP-ABE can be implemented, but KP-ABE is more efficient than CP-ABE. An example of the existing ABE implementing models is Sicari's model (2020) which implements CP-ABE. To overcome the KPI drawbacks of the implemented scheme that are mentioned, the previous middleware needs to be revised and new components need to be added.

The problem is significant to stakeholders, as IoT users experience a noticeable delay when trying to access data. Moreover, security and privacy can be violated if a key is compromised. IoT devices consume more battery power due to higher energy consumption rates in the case of CP-ABE. The problem is also challenging, as there is no prior solution implementing KP-ABE and PRE in NOS.

This thesis proposes a model which prescribes the architecture structure, and functionalities and components of the middleware which can be instantiated into an IoT system. KP-ABE is implemented with PRE in a secure, flexible, and quality-aware IoT middleware known as NOS which was previously constructed by Rizzardi et al. (2016). NOS middleware was chosen because it is the only architecture available in literature that is able to address both security and data quality issues by bringing data processing, privacy, security, and quality closer to the actual data sources, which are crucial factors to any IoT application (Sicari et al, 2016). Additionally, NOS's modular architecture enables developers to extend its functionality and add new features and further extend them in the future. The model proposed by the thesis is original in the aspect of addressing IoT data security and quality issues and extending the functionalities of a previous middleware by implementing KP-ABE and PRE, as there is no previous proposed solution that uses these methods in the middleware. The model is demonstrated using data generated from real IoT nodes from a smart home like weather, temperature, and elecricity. Previous solutions implemented CP-ABE and sticky policy enforcement within the middleware (Sicari et al., 2017; Sicari et al. 2020) but not KP-ABE or PRE. The proposed model improves execution time, CPU load, and memory consumption of IoT devices performing ABE.

1.3 Research Question

The thesis aims to answer the following question:

How can we improve the security and efficiency of ABE in IoT applications?

1.4 Research Goals and Objectives

The thesis research goals are to outline the architecture details and components of an IoT system by constructing a new model artefact which will enhance data security, and improve execution time, CPU load, and memory consumption in IoT applications which perform lightweight operations. The research, therefore, aims to achieve the following objectives:

1. To enhance the efficiency of ABE algorithms in an IoT framework.
2. To enhance the security of the IoT framework by including an efficient key revocation method for ABE.

The proposed DSR artefact is classified as a model that specifies the internal functionalities of the middleware, and the trusted authority. The model proposed by the thesis is original as it extends the functionalities of a previously designed middleware by implementing two new functionalities which are KP-ABE and PRE, thus, prescribing the new components, technical details and requirements needed to implement in an IoT environment.

1.5 Delimitations of the Study

The study aims to achieve research objectives through DSR by using documents as the primary source of data. The documents reviewed were previous research publications, articles, and books. The documents were reviewed to validate and justify the requirements of the IoT middleware including functional and non-functional requirements. The study is independent of any organization utilizing IoT. The reviews and requirements depend on previously designed frameworks and schemes based on ABE for IoT.

Due to the limited resources available, the author implemented and tested the model’s components on a laptop which installs two Ubuntu operating systems. Thus, some entities were emulated on the same device resulting in a single point of failure (SPOF). Another choice was to use a device for each entity; data producer, data consumer, and the trusted authority to demonstrate the artefact and to accurately validate and evaluate it and to prevent having a SPOF.

To obtain a better and more extensive data set of requirements, another choice instead of documents review was to employ survey questionnaires and deploy them among expert IoT users and organizations. However, finding IoT industry experts who are aware of the cybersecurity issues and technical details required a lot of time, and due to the limited amount of time, this strategy was challenging to employ.

2 Extended Background

This chapter provides an extended background and research related to ABE schemes, frameworks, and current IoT and ABE challenges.

2.1 Attribute based encryption

2.1.1 Overview

In recent years, many security protocols have adopted ABE technique as a building block in several distributed environments (Chow, 2016), such as IoT (Sicari et al., 2020), cloud services (Yu et. al, 2010), and medical systems (Ming et al., 2010). ABE is a public key scheme at which both encryption and decryption are based on high-level data access policies. Considering the research problems in IoT scenarios like time efficiency, ABE provides more efficient access control mechanism compared to traditional cryptographic algorithms (Chow, 2016). Moreover, according to Ambrosin, ABE has a lot of advantages when compared to traditional cryptographic algorithms (2016), which include: (1) allowing fine-grained access control based on the recipient attributes, (2) scales independent from the number of authorized users (3) resilient against collusion attacks (4) does not require key sharing or key management algorithms between the participating parties (i.e. data owner does not need to identify destination client)

2.1.2 ABE in IoT

Figure 2-1 shows the typical IoT architecture. In any IoT scenario there exists many data producers, like RFID, actuators, nodes, and sensors, many data consumers and data storage on which data is either temporarily or permanently stored.

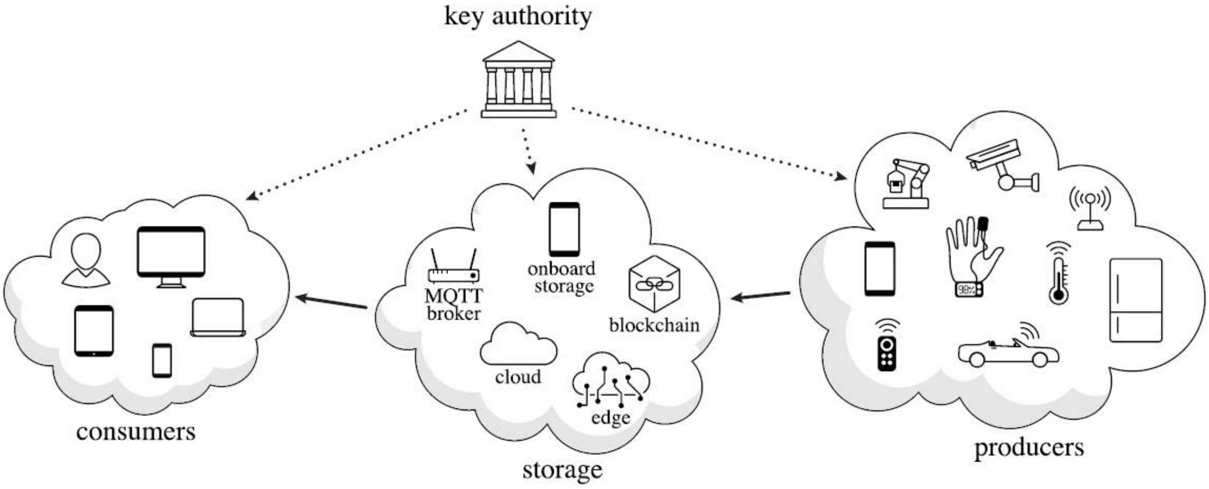


Figure 2-1: Typical IoT architecture from Rasori et. al. (2022)

In IoT, data producers are sensing devices that measure physical quantities or detect events from the surrounding environment. They are constrained, battery powered, and have limited connectivity and computing capacities. While data consumers are devices that display this data to users, or actuators that undertake actions based on this data. They can be tablets, smart watches, smart phones, or even computers with higher connectivity and computing capability than the average data producer. In some scenarios, data consumer devices can be constrained such as actuators. The third entity in the IoT architecture is the data storage entity, this can be implemented in various ways. For example, it can be implemented on the cloud in scenarios where cloud service must be provided to users (Yu et. al., 2010) or on-board in case they store data without transmitting it immediately (Sicari et al., 2020). In each of the cases, data storage can be devices that use IoT communication protocols such as MQTT broker devices, and edge nodes. The typical ABE scheme in literature considers the data storage entity as untrusted. First, on-board storage is considered untrusted due to the physical access to the devices if they are left unattended; they can be easily hacked if the attacker has physical access. Second, cloud storage is considered untrusted because it’s often managed by third parties that can access the data if the encryption scheme is not secure. Thus, it’s essential to encrypt the data when at rest using an efficient, and secure scheme based on ABE. Finally, a fourth entity is required to implement ABE in an IoT architecture, the entity is known as: Trusted key authority (TA). TA is responsible for generating, distributing, and revoking secret keys. These activities are referred to with the general term key management (Rasori et. al., 2022).

2.1.3 Basic ABE algorithms

Attributes are the basic building block in ABE, which is a property associated with a piece of data or a data consumer. An access policy describes the access authorization associated with the data or the consumer. It is represented as a Boolean expression that uses attributes as arguments and represented as a tree in which leaf nodes are attributes, and intermediate nodes are Boolean operators (Rasori et al., 2022). ABE has two types of encryption techniques: 1) key policy attribute-based encryption and 2) ciphertext policy attribute-based encryption. In both techniques, it’s necessary to have a copy of the public parameters which are public and unique for all encrypting parties. While to decrypt data, it’s necessary to own a decryption key that is communicated to the data consumers by the TA (Rasori et al., 2022).

Key policy attribute based encryption

In the KP-ABE scheme firstly proposed by Goyal et al. (2006), the public parameters used for encryption are the attributes, while the decryption keys are associated with the access policies. Access policies describe the “capability to access what”, referred to the owner of the decryption key. Key authorities decide on the access authorization policy when creating the decryption keys and sending them to the data consumers.

According to Goyal’s construction of KP-ABE scheme, it should implement at least the four following algorithms:

1) (*MK*, *EK*) = **Setup**(U). This algorithm takes as input the universe of attributes U and generates a random master key MK, and an associated set of public parameters which define the encryption key EK. The master key should be private and kept secret by the TA. While the encryption key parameter set must be distributed to the data producers. Each attribute *i* is associated to a t_i and a T_i components. Y and y are other public key components.

   $PK = (Y, T_1, T_2, . . ., T_N)$
   $MK = (y, t_1, t_2, . . ., t_N)$

   The setup algorithm is executed by the TA.

2) *(E) =* **Encrypt***(*M, γ, EK*)*. This algorithm encrypts a message M with the attribute set γ and the encryption key EK, it produces the ciphertext E. Each encryption attribute i is associated to an encryption component e_i.

   E = (γ, e', e_1, e_2, . . ., e_N)

   The encryption algorithm is executed by the data producer.

3) (*DK*) = **KeyGen**(MK, τ). This algorithm produces a decryption key DK, which is provided to a data consumer. It takes as input the master key MK and an access policy τ, with access policy attributes i. Each access policy attribute i is associated to a decryption key component dk_i.

   DK = (τ, i, dk_1, dk_2, . . . , dk_N)

   The key generation primitive is executed by the trusted authority.

4) *(M)* = **Decrypt**(E, DK). This algorithm retrieves the message M from the ciphertext E if the access policy embedded in the decryption key is satisfied by means of the encryption attributes γ labeling the ciphertext. The decryption primitive is executed by the data consumer.

Ciphertext policy attribute based encryption

In the CP-ABE first proposed by Bethencourt et al. (2007), the public parameters used for encryption are the access policies, while the decryption keys are associated with the set of attributes. Access policies describe the "capability to be accessed by whom", referred to the encrypted data. Data producers decide on the access authorization policy when encrypting the data.

According to Bethencourt's construction of CP-ABE scheme, it should implement at least the four following algorithms:

1) (*MK*, *EK*) = **Setup**(U). This algorithm acts similarly to KP-ABE setup.

2) *(E) =* **Encrypt***(*M, τ, EK*)*. This algorithm encrypts a message M with the access policy τ and the encryption key EK, it produces the ciphertext E.

3) (DK) = **KeyGen**(MK, γ). This algorithm produces a decryption key DK, which is provided to a data consumer. It takes as input the master key MK and attribute set γ.

4) (M) = **Decrypt**(E, DK). This algorithm retrieves the message M from the ciphertext E if the attribute set embedded in the decryption key is satisfied by means of the encryption attribute policy τ labeling the ciphertext.

CP-ABE vs KP-ABE

Previous research presented performance evaluation of the two ABE schemes KP-ABE and CP-ABE proposed by Goyal et al. (2006) and Bethencourt et al. (2007) respectively. The authors evaluated KPIs such as execution time, data and network overhead, energy consumption, CPU, and memory consumption (Wang et al., 2014; Selar et al., 2017).

The evaluation results show that KP-ABE's execution time of encryption, decryption and key generation operations was less than that of CP-ABE resulting in less energy consumption of devices' batteries. CP-ABE also took more CPU load to perform the operations (Wang et al., 2014). One drawback of implementing KP-ABE scheme illustrated by previous research was that the encryption, decryption, and key generation algorithms were done in a periodic basis, where CP-ABE was more suitable in environments with multiple authorities as encryption, decryption, key generation, and key revocation algorithms were done dynamically (Selar et al., 2017).

However, one of the biggest challenges in IoT is implementing a standardized framework that can be used for all applications (Rizzardi et al., 2016), moreover, the operations of CP-ABE and KP-ABE depend on the IoT application, and are affected by a lot of factors such as data types, the number of attributes, the size of attributes, and the parameters of ABE, to mention some. A more in-depth evaluation and assessment based on the number of attributes and their size is required to show the results on specific IoT applications (Sicari et al., 2020).

2.1.4 Key management

Overview

To be used in practice, ABE schemes should provide some additional functionalities of key management. Specifically, they should provide mechanisms to distribute secret keys (decryption keys), and to revoke them. Distributing the decryption keys to the consumers is an easy task, however, implementing an efficient key revocation mechanism is a challenging task and is an ongoing research effort (Liu et al., 2016; Al-Dahhan et al., 2019). An ABE scheme providing a key revocation mechanism is known as a revocable scheme. Any non-revocable scheme can be extended to a revocable one using the naïve revocation mechanism. In the naïve mechanism, the key authority runs the **Setup** algorithm to create a new master key (MK) and encryption key (EK). After that, for each non revoked consumer, the key authority runs the **KeyGen** algorithm to generate new decryption keys and then distribute them over a secure channel using HTTPS, or simply encrypt the new decryption keys with the consumer public RSA key and send them (Rasori et al., 2022). Key revocation is done when the decryption key is compromised, thus before encrypting any new data, data producers should obtain the new key from the key authority first.

Proxy re-encryption

PRE is a technique that allows an entity to transform a ciphertext which was encrypted using a key to another ciphertext encrypted with a different key without accessing the plaintext itself. By using PRE, one can re-encrypt old ciphertexts to prevent a revoked key from decrypting them.

In this research, the author uses the scheme built by Rasori et al. (2020). Rasori's scheme implements the revocation of a decryption key by means of system-wide update of a subset μ of the access policy attributes to a new version. The subset is called the *Updatee set* and contains the new attributes without which the new decryption key can never be satisfied. Thus, the revoked decryption key will not be able to decrypt any old ciphertext anymore. Updating an attribute i to a new version means that all the cryptographic components in the system related to that attribute are updated. Specifically, for each attribute $i \in \mu$, (1) the t_i of the master key, (2) the T_i of the encryption key (3) the e_i of all the ciphertexts having i as encryption attribute, and (4) the dk_i of all the decryption keys (except the revoked one) having i as access policy attribute are updated to new quantities. The author indicates such new components respectively with t'_i, T'_i, e'_i, and dk' which are computed by the TA. At the time of a revocation, the TA produces a re-encryption key for each attribute in the updatee set then re-encrypts the old encryption components. For simplicity, the author abstracts away the mathematical details of the PRE scheme.

The PRE scheme consists of the following algorithms:

1) (t'_i, T'_i, rk_i) = **UpdateAttribute**(i, MK). It takes as input the master key MK, and it produces the new quantities t'_i, T'_i, and the re-encryption key rk_i. The **UpdateAttribute** algorithm is executed by the TA.
2) e'_i = **UpdateE**(i, e_i, rk_i). This algorithm updates a ciphertext component e_i to a new version e'_i by means of the re-encryption key rk_i. The **UpdateE** primitive is executed by the TA.
3) dk'_i = **UpdateDK**(i, dk_i, rk_i). This algorithm updates a decryption key component dk_i to a new version dk'_i by using the re-encryption key rk_i. The **UpdateDK** primitive is executed by the TA.

2.2 Review of related ABE frameworks and schemes for IoT

The author reviewed articles proposing ABE frameworks and schemes. The articles were filtered in such a way that ABE schemes that perform lightweight operations for IoT applications were included. The author reviewed articles proposing ABE schemes and frameworks in IoT applied in the scenarios of most interest to IoT users, which are discussed, such as healthcare, home, and industrial applications. The selection by filtering the newest ABE schemes and frameworks that are considered state-of-the-art and were reviewed to discover the challenges of the related IoT frameworks used in the scenarios.

In the article written by Sicari et al. (2020), the authors proposed an architecture of a framework using the previously designed middleware for encrypting IoT data using CP-ABE. The paper represents the architecture of the IoT system that implements CP-ABE within the NOS middleware. The architecture consists of four entities: data producer (data source), data consumer, a trusted authority, and the NOS middleware, as shown in figure 2-2. As per Sicari's implementation (2020), data is first produced by the data source, then encrypted by the public RSA key of the data producer and then sent to the NOS middleware. In the NOS middleware, data is decrypted and then stored as *raw data*, analyzed for security purposes, then encrypted using CP-ABE, using the policies which are created in the NOS middleware. The trusted authority communicates with the NOS middleware to send the encryption key that is used to encrypt IoT data using ABE. Users, then, can access the data by means of fine-grained access according to the attributes they possess and decrypt the data using the secret decryption key sent to them by the TA.

In another article written by Yao et al. (2015), the authors proposed a lightweight ABE scheme that uses elliptic curve cryptography (ECC) instead of bilinear pairing used in most ABE schemes. The authors believed that it is difficult to implement bilinear-pairing-based algorithms in resource-constrained IoT devices and proposed a new KP-ABE scheme which implements the four algorithms based on ECC and its related primitives such as Lagrange secret sharing. Operations in ECC consist of basic prime field operations and point operations. The former operations are simple; however, the latter operation refers to point scalar multiplication, which is refined by point add and point double operations (Yao et al., 2015).

Another article written by Sanchez et al. (2014) introduced a wireless body area network (WBAN) architecture based on a publish-subscribe messaging paradigm for wearable sensors and devices in eHealth domain. WBAN shares messages to users through fine-grained access protocol based on a CP-ABE scheme that is previously introduced by Guo et al. (2014). In their scheme, Guo et al. proposed a new algorithm for keeping the size of decryption keys constant by using AND-only tree access structure (2014). In Sanchez et al.'s architecture, WBAN is composed of several heterogenous devices implanted in a patient's body that provides physical and physiological parameters of the patient, such as temperature, heart rate, and position (Sanchez et al., 2014). Each BAN node is configured with a policy service that determines which entity can access the data, this policy may be fixed (e.g., Only a nurse or a doctor can access the data published by a sensor) or may depend on the context (e.g., location, readings of other sensors, state of the patient) (Sanchez et al., 2014). Each node that wants to join the WBAN obtains the secret key (decryption key) and public parameters (i.e., attributes) from the key generation center (KGC), after this each node can subscribe and publish its content on the API provided in the architecture by encrypting it using the appropriate policies (Sanchez et al., 2014)

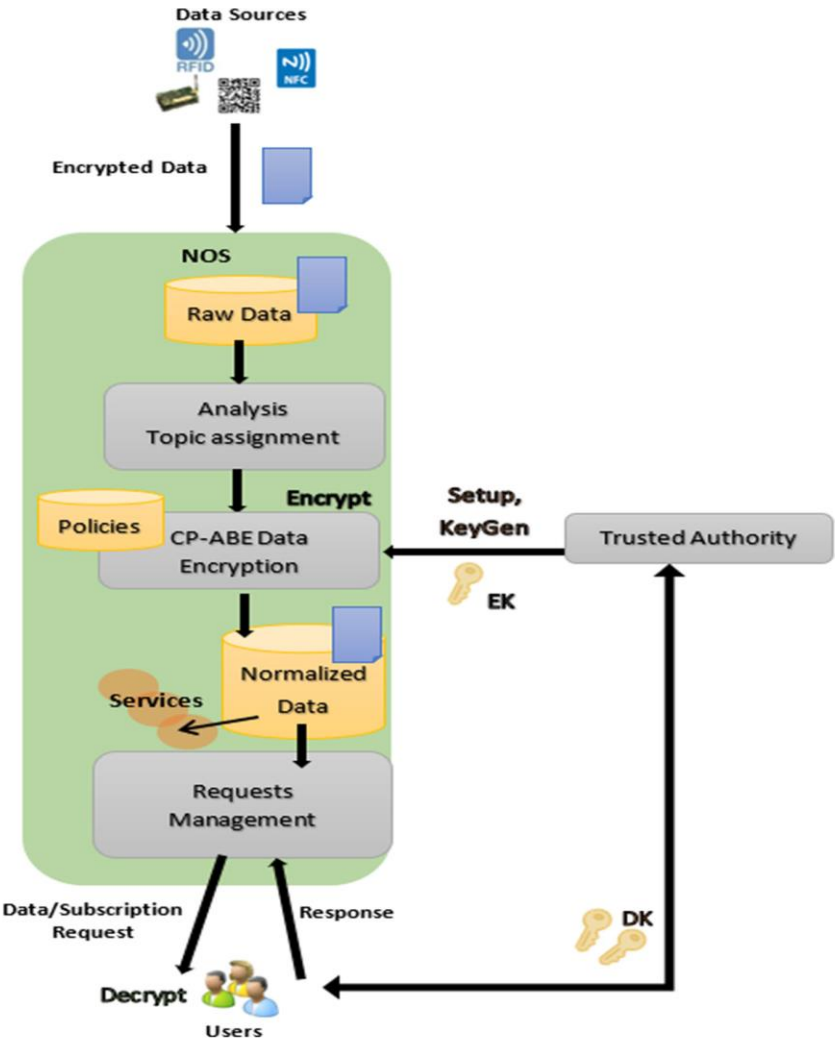


Figure 2-2: Proposed NOS architecture by Sicari et al. (2020)

In another article published by Singh et al. (2015), the authors proposed an architecture of a system that adds on the functionality of MQTT protocol (based on TCP) which is used for communication between IoT devices (Locke, 2010). Singh proposed a secure MQTT (SMQTT) framework that encrypts data published and subscribed to users by implementing KP-ABE and CP-ABE. In the proposed architecture, Singh et al. implemented the ABE techniques into the MQTT broker. The broker's main functionalities were to: (1) generate the decryption keys (2) distribute them to the users through MQTT protocol, (3) act as a framework for storing messages (Singh et al. 2015). According to Singh et al., each entity should register itself with an identity and a set of attributes with the broker. The broker then sends the public parameters to the registerers to encrypt the messages. After that, the sender sends an MQTT message *Spublish* after encrypting it using the encryption key and the public parameters as shown in figure 2-3.

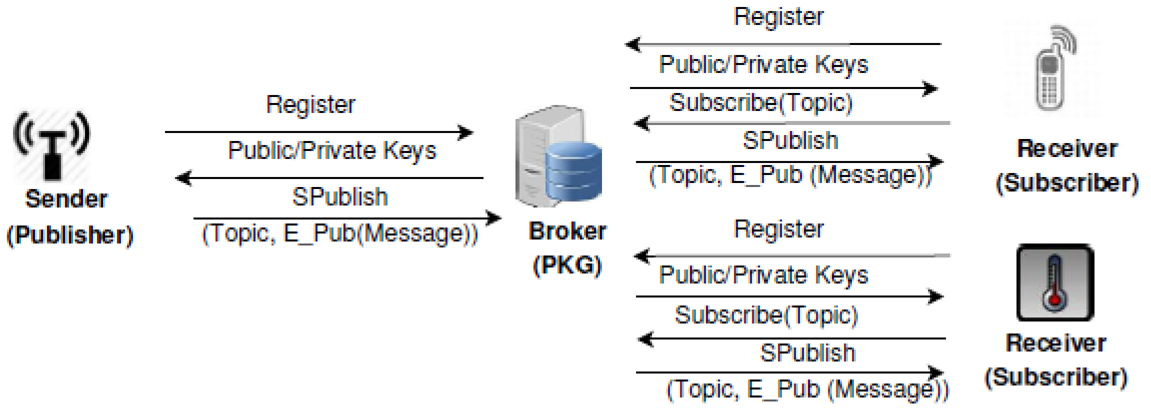


Figure 2-3: Proposed SMQTT architecture by Singh et al. (2015)

2.3 Limitations of the related data sharing frameworks and schemes

The model proposed by Sicari et al. (2020) employs CP-ABE for encrypting IoT data. While this mechanism was proved to be more efficient than sticky policies in the storage requirements of the NOS middleware (ibid.), it requires more CPU load, and more time to encrypt and decrypt the data and requires more time to generate the decryption key than the time required by both sticky policies (Sicari et al., 2020; Selar G. et al., 2017). Moreover, the mechanism described in the paper uses CP-ABE which does not provide for efficient re-encryption mechanism in the case of key revocation. On the other hand, this thesis's model implements KP-ABE scheme which is proved to be faster than CP-ABE in encryption, decryption, and key generation as evaluated by previous research (Wang et. al., 2014; Selar G. et al., 2017). Furthermore, the proposed model of the IoT system is original as it introduces new functionalities to NOS middleware that are required to implement KP-ABE with PRE which was not deployed or tested before in the middleware. KP-ABE improves efficiency KPIs and PRE enhances the security and privacy of the data by preventing unauthorized access to it when the key is compromised.

The authors of the lightweight ABE scheme in Yao et al.'s publication (2015) performed a thorough evaluation on the computational and communicational overhead of the proposed scheme, although they proved that its more efficient than other existing KP-ABE and CP-ABE schemes, their proposed scheme is not flexible in revoking decryption keys. The secret sharing scheme used to generate the key made it difficult to revoke decryption keys flexibly (Yao et al., 2015). Comparatively, the study's proposed model combines KP-ABE with a flexible key revocation mechanism.

The WBAN architecture proposed by Sanchez et al. (2014) employs a lightweight CP-ABE scheme that is suitable to be implemented in lightweight devices like small sensors known as implantable medical devices (IMD). However, to obtain the lightweight feature of the scheme, the authors had to construct an algorithm that uses only AND access gates. This restricts the types of policies used in different IoT applications and services and limits the access control flexibility (Sanchez et al, 2014). On the contrary, this thesis implementation is based on Goyal et al.'s scheme (2006), that offers a far greater degree of expressiveness for ABE access policies.

Finally, Singh et al.'s model (2015) adds new features on the functionality of MQTT protocol and uses a broker to act as a trusted key authority as well as a communication enabling entity that stores messages. Thus, the system is considered a SPOF as it combines two functionalities on one device. This results in a risk as one malfunction can cause the entire system to stop operating. On the other hand, the thesis model uses two separate entities; one for representing a TA, and another for representing the NOS middleware to avoid having a SPOF. Moreover, using the NOS middleware combined quality, privacy and security for the stored data that allows the users who access the data to be aware of the levels of reliability and trustworthiness of the services gathered by the NOS (Sicari et al., 2020). Furthermore, Singh et al.'s system used an ABE encryption technique that does not provide re-encryption during key revocation. In contrast, the thesis' proposed model includes an efficient lightweight key revocation mechanism. Table 2. lists the existing frameworks along with the encryption type, pros, and cons discussed in sections 2.2 and 2.3.

Framework/ scheme	**Encryption type**	**Suitable Scenarios**	**Pros**	**Cons**	**Reference**
Previously designed NOS middleware	CP-ABE	Home, Business Activities Analysis, Retail, Healthcare	• Lightweight as it does not implement re-encryption mechanism • No storage requirements needed for re-encryption	• No efficient revocation mechanism • High CPU load • High memory consumption • Long data retrieval delay	(Sicari et al., 2020)
ECC-lightweight elliptic curve cryptography scheme	KP-ABE	Home, Healthcare, Human wearables	• Less expensive than schemes based on bilinear pairing, which makes it lighter	• No efficient key revocation mechanism	(Yao et al., 2015)
WBAN architecture	CP-ABE	Healthcare, Human wearables and sensors networks	• Suitable for eHealth domain applications which use lightweight implementations	• No flexible access control policy • No efficient key revocation mechanism	(Sanchez et al., 2014)

SMQTT with ABE architecture	KP-ABE and CP-ABE	Social networks, Vehicle to Vehicle communication (V2V), and Sensor networks	• Less devices are needed, as only a broker is used for representing a TA and a data storage unit	• The architecture provides a SPOF. • No efficient key revocation mechanism	(Singh et al., 2015)

Table 2-1: Overview and summary of related data sharing frameworks and schemes based on ABE

2.4 Summary

Sections 2.2 and 2.3 provided a thorough overview for ABE frameworks and schemes used in healthcare, home, and industrial applications that showed that the frameworks used do not combine security, privacy, data quality and efficient ABE schemes in a framework that is state-of-the-art like the one used in the research study. Combining all these factors together is the aim of this thesis to achieve the objectives of the research. The thesis work is original in the sense that it extends the functionality of NOS middleware by using an efficient KP-ABE scheme along with an efficient key revocation mechanism. The thesis solves the problem of inefficient key revocation, time delay, and CPU load stated by the previous case study (Sicari et al., 2020). Moreover, the thesis model uses a KP-ABE scheme that offers a high degree of policy expressiveness and solves the problem of limited access control flexibility experienced by Sanchez et al. (2014). The thesis proposes a DSR artefact that is a prescriptive model representing the architecture of the IoT system including the data storage components, encryption modules (i.e., KP-ABE and PRE), automation scripts, and TA components required to produce an instantiation and used in an IoT environment.

3 Research Methodology

This chapter discusses the methodology and research strategies, data collection methods and data analysis methods used to conduct the thesis.

3.1 Research strategy and methods

3.1.1 Overview of design science research process

The thesis facilitates the IoT data sharing framework architecture using the DSR framework presented by Johannesen and Perjons (2014). According to Johanneson and Perjons, DSR consists of five main activities ranging from problem investigation and requirements definition, through artefact design and development, to demonstration and evaluation as shown in figure 3-1 (2014). After evaluating the artefact, research results can be presented in many different structures, but the most common structure for presenting DSR is the IMRAD structure which consists of introduction, methodology, results, analysis, and discussion sections (Johannesen and Perjons, 2014). The thesis uses a modified IMRAD structure by adding the extended background section.

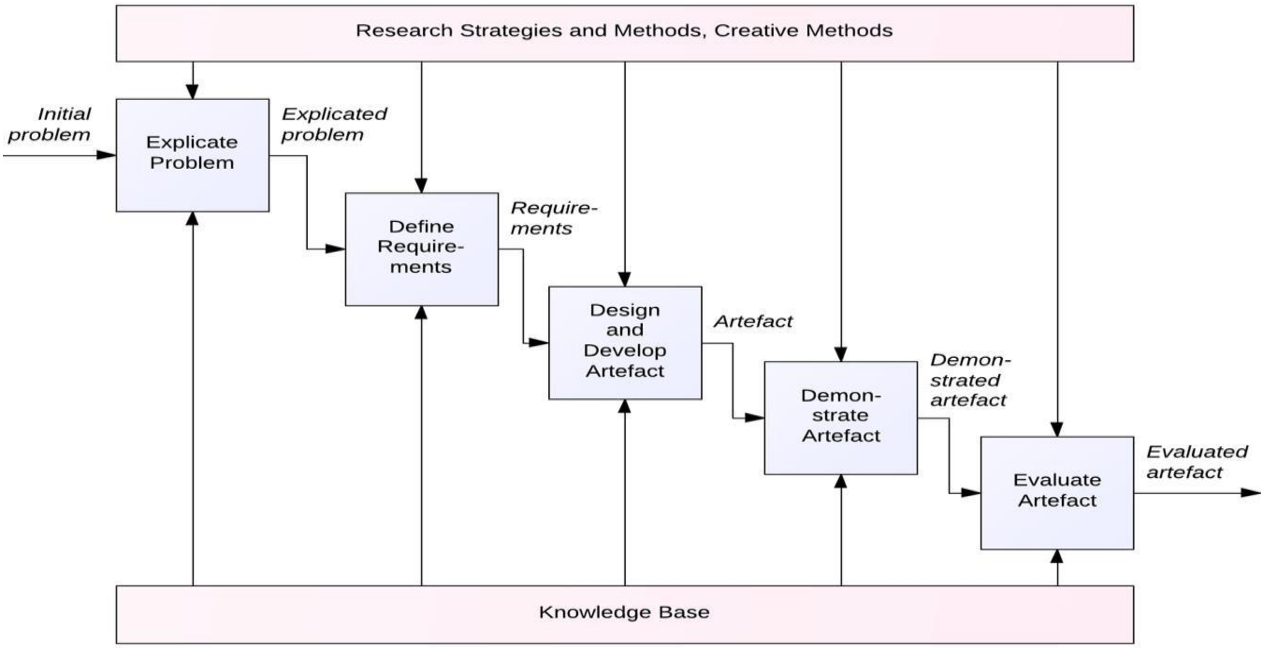


Figure 3-1: An overview of design science research process from Johannesson and Perjons (2012, p.33)

3.1.2 Data collection methods

According to Johannesson and Perjons, different scientific communities have evolved and established several data collection methods for empirical research to support researchers in creating and presenting their research results, which are also useful for DSR when investigating practical problems, defining requirements, and evaluating artefacts (2014). This thesis employs document review and case study for the data collection method in the first three steps: problem explication, requirement identification, and artefact design.

Case study

A case study uses one instance of a phenomenon as a reference and investigates it, then it offers a rich, in-depth description and insight of that instance to the audience (Johannesson and Perjons, 2014). The author investigated a previous phenomenon where a research gap was found and used this case study as a reference to explicate the problem, gather information on building the framework, and identify the requirements. Investigating a case study provided the author with deep knowledge about the phenomenon and deep insight into the problem that is to be solved in that phenomenon, which makes the solution perfectly tailored for this problem. However, Johannessen points out that a common criticism of case studies is that it is only applied to the instance being studied and that the results cannot be generalized (2014).

Document review

A document review is a descriptive or analytic summary of material related to a study's area (Hannah, 2019). The author performed a document review of existing cryptographic schemes and architectures that store ABE-encrypted IoT data to find out the needs of IoT applications and the limitations of the data sharing frameworks. In other words, the author found the requirements using the knowledge acquired from previous research documents of ABE and IoT. Using documents related to the thesis topic as a data source is a time and cost-effective data collection method. However, it's difficult to judge the credibility of these documents. To overcome this problem, the author based most of the research on accurate databases like Google Scholar, IEEE Explore, Springer Link, and Science Direct.

Alternative data collection methods

Questionnaires

Another effective way to collect information is questionnaires. They consist of multiple questions which gather the opinions, and perception of participants of a certain field. This data collection method is quick, and questionnaires can be tailored then distributed to different stakeholders from different business sectors to gain insight into different business objectives. However, the author did not use questionnaires as they can be a difficult approach to gain a high response rate, as respondents can ignore the request to answer a questionnaire, thus, the required amount of knowledge will not be collected from this approach.

Interviews

Interviews consist of three types: structured, semi-structured, and non-structured. Structured interviews are interviews that are pre-formulated with questions that are standard and are asked in a specific order. Thus, they do not provide deep insight into the problem. Semi-structured interviews contain questions that are standard but are not ordered in a specific order. While non-structured interviews are flexible, and the questions are not standard. Interviews were not the choice of data collection method in this thesis because they are time-consuming. Interviews require a lot of preparations in advance, then another process after collecting the data which is transcription and data analysis. Therefore, it is a very long process and requires a lot of time, unlike case studies and document reviews which are quick and convenient.

3.2 Justification for design science research

According to Johannessen and Perjons (2014), empirical research strategies such as experiments, and mathematical proofs describe, predict, and explain the world. In contrast to empirical research, DSR is not content to just describe, predict, and explain but also to change the world, improve it, and create new worlds. The primary reason for choosing DSR and not empirical strategies is that it proposes artefacts that can help people fulfil their needs, overcome their problems, and grasp new opportunities.

Experiments strategy could have been applied in the research project like it was applied before in Sicari et al.'s (2020) research. An experiment is an empirical study that investigates cause and effect relationships, proves, and disproves them by alternating the values of independent variables, and reflects on the values of the dependent variables (Johannesson and Perjons, 2014). Sicari's main aim was to compare the performance evaluation of two encryption methods. However, reflecting on the dependent variables only is not the purpose of this research project. The purpose of the thesis is to apply a strategy that constructs useful solutions that include tools, frameworks, methods, procedures, models, and techniques not only by reflecting on the dependent variables but also by investigating the functional, and non-functional requirements of the new tool, a step before developing it, thus making use of the steps of DSR process to create a useful solution, demonstrate the solution then evaluate it (Peffers et al., 2007).

In another research made by Yao et al. (2015), mathematical proof strategy was applied. Yao et al. constructed a new scheme for ABE encryption based on ECC algorithms. The mathematics of the algorithms' schemes proved that the new scheme was more efficient in execution time theoretically. Mathematical proof alone is not enough because the aim of the thesis is to demonstrate the required artifact in an IoT scenario, then evaluate the functional features of the designed artifact. Another reason for favoring DSR over mathematical proof is that DSR produces a range of artifacts, such as instantiations, models, constructs, and methods which are designed after explicating the problem, and investigating the functional and non-functional requirements.

DSR communicates the solutions that help serve the audience in a clear and effective way. As Hevner et. al. explains, it involves a rigorous process to design artifacts that solve observed problems, make research contributions, evaluate the designs, and communicate the results to appropriate audiences (2004). After that, the process can be shared with the target audience such as academia, and IT industries through the research paper.

3.3 Application of design science research

DSR consists of five main activities ranging from problem explication and requirements definition, through artefact design and development, and finally to demonstration and evaluation. The author's application of design science research strategy is explained below.

3.3.1 Explicate the problem

The author performed a document review of existing IoT architectures and schemes for data encryption based on ABE to explicate the problem and identify the requirements. A previous case study was reviewed, and some research gaps were identified, like inefficient ABE operations, the need for further evaluation, and insecure key revocation. Therefore, the problem explication was made through a case study and a literature study.

3.3.2 Define requirements

For defining the requirements of the architectural model, the author reviewed existing models and schemes for IoT applications based on ABE. A requirement is a property of an artefact that is needed or desired by stakeholders in the practice which may guide the design and development of the artefact (Johannesson and Perjons, 2014). A functional requirement specifies that an artefact should offer some function and provide some benefits for the users, on the other hand, a non-functional requirement specifies in what way the artefact shall provide its functions to users (Johannesson and Perjons, 2014). The requirements definition was done through a document review as the primary knowledge source. Through the document review, the author was able to identify both functional and non-functional requirements of the artefact which are required to improve the previous IoT model and are discussed in section (2) of chapter (4).

3.3.3 Design artifact

The author used a design thinking approach to design the artefact and chose the technical functionalities required to implement in NOS middleware from the knowledge base acquired from reviewing existing ABE schemes. The artefact design details are discussed in section (3) of chapter (4).

3.3.4 Demonstrate artifact

The artefact was demonstrated by implementing the NOS middleware on Ubuntu OS that emulates the Raspberry Pi and deploying the TA on another Ubuntu OS together with the data consumers and producer. The NOS analysis, encryption and storage processes were tested and executed in a computer simulation using the laptop to test the feasibility and the functionality of the different components of the proposed model. The demonstration is discussed in section (4) of chapter (4).

3.3.5 Evaluate artifact

The author applied an ex-ante approach with artificial strategy to evaluate the artifact which is computer simulation. An artificial evaluation refers to an evaluation at which the artefact is evaluated in an artificial setting, ex-ante evaluation evaluates the artefact before being implemented and is the most suitable evaluation technique used to evaluate a prototype or a design (Johannesson and Perjons, 2014). The evaluation results are discussed in section (5) of chapter (4)

Table 3-1 shows the DSR canvas.

Design science canvas

Practice

- Practice: data transfer and storage in IoT applications
- Purpose: constructing an architecture of an IoT system that uses an efficient, secure, and flexible middleware to enhance the security and privacy of data in IoT applications and provides a lightweight ABE scheme that suits the applications of lightweight IoT operations
- Activities include data encryption, decryption, key generation, data storage, data transfer, key revocation based on proxy re-encryption
- Stakeholders: data users in IoT smart home scenario

Problem

- Broad problem: the current data sharing frameworks and schemes based on ABE are not combining security, quality, efficient KPIs, and proxy re-encryption method
- Precise problem: A previous ABE architecture requires much CPU load, consume more energy, and take a long time to encrypt, decrypt, and generate secret keys and has no efficient revocation mechanism
- The problem is significant to IoT users, as their data may be vulnerable to attacks, and they experience a delay while trying to retrieve data. The problem is challenging because no solution addressed data quality, security, and efficiency

Research Process

- An extensive case study and literature review was conducted on existing ABE schemes and architectures for problem explication
- The author performed a document review on IoT security requirements and existing ABE schemes and architectures to define requirements
- The author used design thinking to define, and test the solution for design and development
- For the evaluation, the author applied ex-ante evaluation with artificial approach

Artefact

The artefact is identified as a model. The thesis’s output is a model of an architecture used in IoT applications for data sharing based on ABE. The model consists of four entities which are: data producers, data consumers, NOS middleware, and the TA. Data producers send data to the NOS middleware, where it is processed and stored. The artefact extended the functionalities of the NOS middleware by implementing KP-ABE and PRE method. Data consumers access the data based on the access policies issued from the TA. The TA generates encryption keys, sends them to NOS and generates decryption keys and distribute them to users

Requirements

Functional Requirements

- The artefact should be able to improve data security, and quality through adding useful metadata regarding security, such as level of confidentiality, integrity, privacy and data quality like such as level of accuracy and precision
- The artefact should add a layer of security by applying KP-ABE
- The artefact should provide an efficient key revocation mechanism

Non-functional Requirements

- The artefact should be effective
- The artefact should be modular
- The artefact should be ethical

Quality and Effects

Quality

- The artefact fulfills all functional and non-functional requirements
- The model is simple and easy to use for protecting IoT data in transit and storage

Effects

- The artefact reduced the data retrieval delay
- The artefact improved IoT data security, quality, and privacy
- The artefact introduced a key revocation mechanism which improved the efficiency of re-encryption during key revocation

Knowledge Base

Previous literature was used to acquire knowledge about IoT frameworks, ABE schemes, and revocation mechanisms.

Table 3-1: Design Science Canvas

3.4 Research ethics

For this thesis, the author employed DSR, which considers the potential ethical and societal consequences of using the artefacts they produce (Johannesson and Perjons, 2014). To capture these requirements, the author followed the principles suggested by Myers et al. (2014). The principles are (1) *The public intere*st: In this thesis, the author identified all the stakeholders who may be affected by the artefact and considers what benefit or harm may result from using them, (2) *Privacy*: The author ensured that adequate safeguards are in place to protect privacy of people who are directly involved with the current project and also those who might use the artefact in the future, (3) *Honesty and Accuracy:* The author explicitly reported the findings of the artefact to the readers.

4 Results and Analysis

This chapter discusses the application of design science research method and presents the results of the study obtained by following the design science research steps.

4.1 Explicate the problem

In this phase, the author explicated the problem from a case study and applied document review method as a data collection method to explicate the research problem required to be solved.

Security, privacy, and data quality represent critical requirements to IoT services without which large scale adoption of IoT services can never be maintained. To ensure that these requirements are met, IoT designers must ensure that they control source validity and information accuracy (Guo et al., 2013). Furthermore, timeliness and data accuracy need to be considered to increase the data quality (Metzger et al., 2012). As denoted in the document review of the previous ABE frameworks used in different scenarios, no previous framework combined all the necessary requirements together. In the case study reviewed by this article, the framework proposed by Sicari et al. (2016) combined privacy and data quality but the issue it did not solve was the timeliness of the data retrieval using the implemented ABE encryption method. Another issue is that the proposed framework lacks an efficient and effective key revocation method that ensures security and efficiency. This led to the following research question:

How can we improve the security and efficiency of ABE in IoT applications?

The subquestions to the research question are:

- **RQ1:** How can the efficiency of ABE operations in the previous model be enhanced?
- **RQ2:** What is the effect using PRE in the model in terms of efficiency and security?

To solve these issues, the author constructed a model of an IoT security architecture that prescribes the components and functionalities of the IoT system. The IoT security architecture ensures data quality, privacy, and security by combining analysis modules, an efficient KP-ABE scheme, and PRE method for enhancing the efficiency and the security of the ABE operations.

4.2 Requirements

The second stage in DSR is requirements definition. In this stage, the author defined the functional and non-functional requirements to the artefact which were gathered from previous research documents proposing similar solutions implementing ABE in IoT frameworks. The author defined the requirements needed to ensure the artefact offers security, privacy, quality, and efficiency in IoT applications.

4.2.1 Functional Requirements

In DSR, functional requirements explain what the artefact should do for the users and what benefits to provide to the users (Johannesson and Perjons, 2012). The functional requirements that should be met in the model are discussed below.

- The artefact should enhance data security, privacy and quality through analysis and data annotation. The idea of the artefact to perform security, and quality analysis are adapted from the system architecture proposed by Rizzardi et al. (2015) which is based on the concept of NOS that creates a distributed storage for handling IoT data acquired from different heterogenous sources. The NOS model uses several modules that process the data before storage. Data is initially analyzed in terms of security then analyzed in terms of quality by assigning scores to authentication, confidentiality, integrity, and privacy based on the data source credibility and encryption method.

  Rationale: The proposed model uses a middleware that uses analysis and data annotation modules to provide a score of data security, privacy, and quality to IoT users. This score is then used by the application users to specify the minimal level of security, privacy, and quality suitable for their own purposes.

- The model should add a layer of security by applying KP-ABE. The model uses an encryption module that implements the KP-ABE scheme constructed by Goyal et al. (2006). The idea of applying KP-ABE as a functional requirement instead of CP-ABE was adapted from reviewing the performance evaluation of KP-ABE done by Wang et al. (2014) and from the research gap of Sicari et al.’s (2020) research. The performance evaluation of KP-ABE showed that the four encryption operations: key setup, key generation, encryption, and decryption are faster than the operations of CP-ABE scheme. Furthermore, the operations took less CPU load, and the used devices consumed less energy compared to CP-ABE.

  Rationale: The proposed model uses KP-ABE encryption after data analysis and annotations to encrypt data using fine-grained access control using a KP-ABE scheme which is more efficient in performing ABE computations.

- The model should provide a key revocation mechanism based on PRE. The requirement for the model to offer PRE is adapted from the security model constructed by Yu et al. (2010) to enhance ABE key revocation methods. The authors of the model argue that the proposed PRE scheme is efficient, lightweight, and suitable for most IoT applications which require battery-constrained devices and lightweight operations.

  Rationale: The proposed model adds a functionality of key revocation based on PRE so that ciphertexts are re-encrypted when the key is compromised so that the compromised keys won’t be used to decrypt old data. New keys are generated to encrypt new data, thus enhancing the security of the IoT application.

- The model should enhance lightweight features of ABE operations by delegating most of the burdensome operations to the TA and maintain only efficient and light operations. The idea of the model to enhance efficiency is adapted from the research problems identified by previous research (Wang et al., 2014; Sicari et al., 2020; Yu et al., 2010). Research indicates that IoT nodes cannot implement complex operations as they are resource, and battery-constrained so heavy operations cannot be implemented on IoT devices but should be delegated to a more resourceful device.

  Rationale: The proposed model will maintain the lightweight feature of ABE operations in the middleware by executing light and efficient operations, and delegate heavy computations which require a lot of energy and time to a more resourceful device.

4.2.2 Non-functional requirements

The non-functional requirements are discussed below.

- The artefact should be modular. The idea for the artefact to have modularity as a requirement comes from the fact that modularity helps in enhancing the current feature set by adding new features that suit the applications where the system will be implemented in the future. Moreover, the idea comes from the concept of ‘Modular Artefact’ where different modules implementing functions can be integrated together in a distributed system. (Ngo et al., 2004)

  Rationale: The proposed model provides a modular structure that implements different security specifications. This also allows the model to be extended with extra functionalities in future applications.

- The artefact should be effective. The idea for the artefact to be effective comes from the 5Es framework described by Johannessen and Perjons (2014). The artefact should produce desirable effect in IoT practices and achieve the research goals by providing efficiency, and security (Rizzardi et al., 2015).

  Rationale: The proposed model shall guarantee to achieve security goals such as privacy, quality, and integrity and enhance efficiency in IoT applications.

- The artefact should be ethical. This requirement derives from the current security and privacy issues experienced by IoT services and applications discussed in chapter (1). Ethicality imposes that the model shall provide data confidentiality, integrity, and availability. It also imposes that the model shall provide IoT security goals of privacy, and data quality.

  Rationale: The model shall adhere to ethical norms by promoting security goals such as the CIA triad. Since IoT applications nowadays face a lot of security and privacy challenges, the model shall handle IoT data with carefulness and respect to the data handling process.

4.3 Develop Artefact

According to DSR artefacts presented by Johannesen and Perjons (2014), the thesis’ artefact is classified as a model. The project's output is a model of an IoT security architecture that is based on ABE and can be used to construct an instantiation and implemented in several IoT applications like home, business analysis, retail, and healthcare. The components that the are prescribed by the model and are chosen based on the requirements of IoT applications elicited from previous research. The main factor of selecting the components are efficiency and security, which were indicated as research requirements in the reviewed architectures and in the previous case study conducted by Sicari et al. (2020).

According to the architectures of IoT, the artefact’s architecture is composed of three entities: (1) The data storage and processing entity which is the NOS middleware that fetches, processes, stores, and encrypts data, (2) The TA that handles requests, generates keys, and distributes keys to NOS and the consumers, (3) The data consumers that request data, receives decryption keys and decrypts data.

4.3.1 NOS Middleware

Step 1: Data security and quality analysis

In the first step, data is sent by the data producer to NOS using HTTPS protocol where the first destination of the data is the *Raw data* collection where three analysis modules are applied to analyze data in terms of security and data quality.

The rationale of analyzing data in terms of security and data quality

As IoT applications' scenarios are different, the choice to provide a score of data quality and security makes the solution flexible for smart integration in different application scenarios. Security, privacy, and data quality present critical requirements for every IoT scenario. Assessing data source quality and security and assigning scores can validate information accuracy and credibility, also previous authors argue that score assignment can be the best solution to filter data based on a minimal score of data security and quality that is required in each application (Rizzardi et al., 2015).

How data security and quality modules work in the IoT security model

The process starts with NOS and the data sources, where NOS pulls the data stored by the data sources from the local host server database *Sources* database. The data transmission in this phase is secure and is done over HTTPS, the data's destination is a distributed cloud storage unit: *Raw data* which is deployed on MongoDB, where it is stored there for analysis. MongoDB was used to deploy the databases due to its high availability offering users fast and easy access. Three analysis modules are used to perform *Source*, *Security,* and *Quality Analysis.* The steps of evaluating data security and quality are shown in figure 4-1.

Quality and security are evaluated based on the type of encryption algorithm and the score assigned to it by the TA. NOS fetches data from *Sources* collection then stores data into *Raw data* collection, which is located on the local host server. Privacy and confidentiality are evaluated by assigning a score to the encryption schema (based on the robustness of the used encryption technique and the adopted key distribution schema) which is decided by the system administrator, Integrity is then checked, and a score is given (0 for a violated and 2 for an unviolated data), authentication is set to 1. For non-registered sources confidentiality, privacy and authentication are set to 0, while integrity is set to 1. Completeness, and accuracy are set by the TA based on the number of collected versus expected values, and the difference between the mean of values and a reference value. The purpose of this step is for the TA to set a minimal standard for the quality and security of data based on the required security levels, so users can access only data of highest security and quality

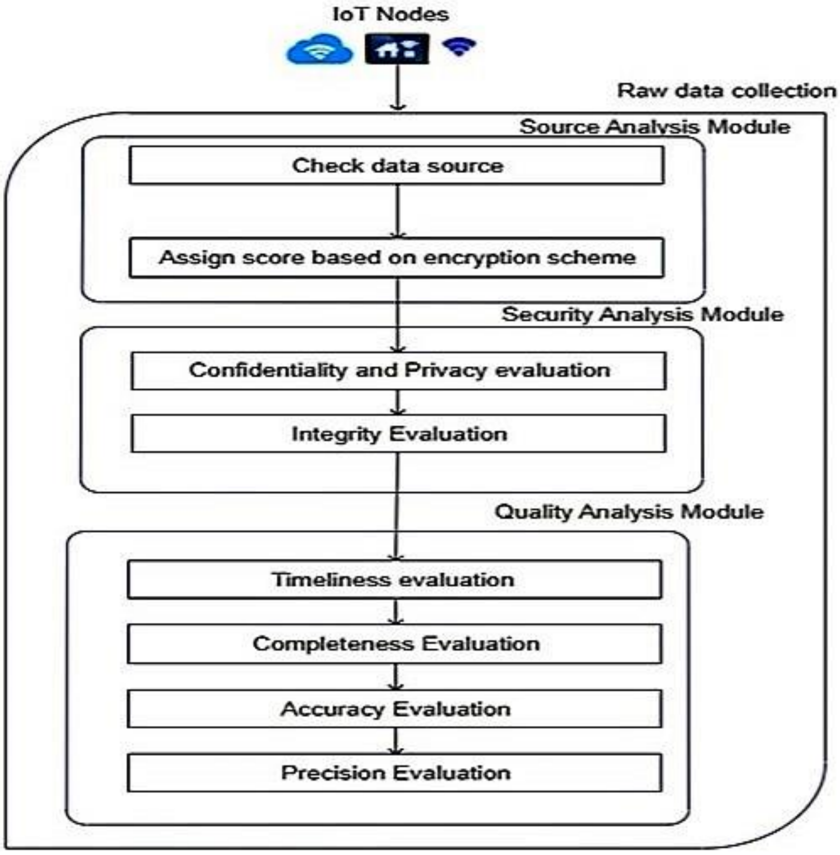


Figure 4-1: Functions of NOS analysis modules

After setting the values, the data are annotated with a set of metadata (a score for each security and quality level) and stored in *Raw data* collection.

Step 2: KP-ABE and PRE

The second step involves KP-ABE encryption of *Raw data*. The process starts within the NOS, NOS fetches the data from the *Raw data* storage unit and encrypts it using the KP-ABE and PRE encryption module and exports it to the *Normalized* storage unit.

The rationale of utilizing KP-ABE and PRE modules to encrypt data and revoke keys

To achieve secure, scalable, and fine-grained access control on the outsourced IoT data, the author utilized and uniquely combined KP-ABE with PRE in the proposed model. Previous case study tested by Sicari et al. (2020) revoked keys by executing the key setup and key generation primitives all over again resulting in many problems which are: (i) previous ciphertexts can still be accessible by attackers who compromised the keys, (ii) the execution time needed is a lot and the operation is not efficient. To solve this issue, the author uses the PRE scheme constructed by Yu et al. (2010) due to its scalability, efficiency, and to enhance the security of the proposed model.

The benefits of combining KP-ABE with PRE are efficiency, and security compared to the previous case study documented by Sicari et al. (2020).

How KP-ABE and PRE modules work in the IoT security model

The data encryption is performed using the attributes stored in the *Attributes* data storage unit. The steps of KP-ABE are described below.

1. **System setup**

This outputs the system public parameter (EK) and master key (MK). The TA is responsible for calling the key-setup function presented in section 2.1.3. Then, the TA creates the attributes version list (AVL) by inserting all the attributes used in the system for each dataset along with their version. The TA signs EK and AVL and sends them to the NOS framework through HTTPS.

2. **KP-ABE encryption**

This step involves data encryption using KP-ABE. NOS is responsible for performing: *KP-ABE-Encrypt* algorithm of the KP-ABE scheme presented in section 2.1.3.

To include the new components/functionalities required by the KP-ABE scheme in the proposed architecture, the previous NOS architecture must be revised. The KP-ABE data encryption module has three main goals: (i) it performs encryption in place of data producers; thus, it maintains the lightweight property of IoT nodes and prevents them from executing complex and expensive tasks, (ii) it properly associates the data with the latest version of attributes stored in AVL, (iii) it then stores normalized data in the *Normalized Data* storage unit in encrypted form, so data are protected in case NOS is compromised by unauthorized parties.

3. **KP-ABE proxy re-encryption**

This step is executed by NOS in case of key compromise or user revocation involves as it involves the re-encryption of old ciphertexts to prevent their compromise. NOS uses the latest PRE key sent by the TA to encrypt the resulted KP-ABE ciphertext, the old ciphertexts components will be updated by calling *UpdateE* function, then associates it with its related attributes and store it in *Normalized* database.

The sequence diagram of data analysis, encryption and storage is shown in figure 4-2.

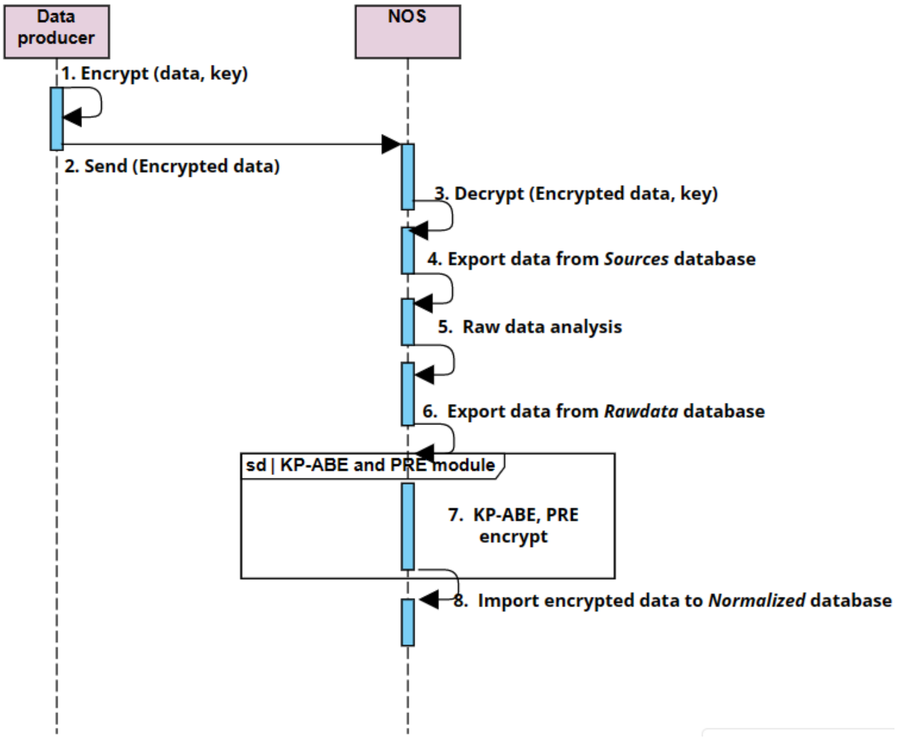

Data producer
NOS
1. Encrypt (data, key)
2. Send (Encrypted data)
3. Decrypt (Encrypted data, key)
4. Export data from *Sources* database
5. Raw data analysis
6. Export data from *Rawdata* database
sd | KP-ABE and PRE module
7. KP-ABE, PRE encrypt
8. Import encrypted data to *Normalized* database

Figure 4-2: Sequence diagram of NOS analysis, encryption, and storage

The new architecture of the NOS middleware combining the new components is shown in figure 4-3.

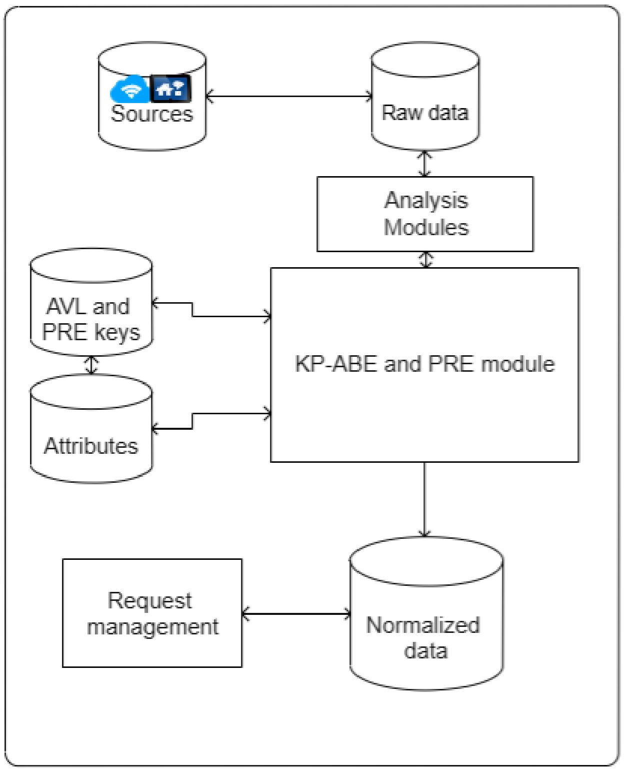


Figure 4-3: NOS middleware architecture

4.3.2 The TA

The TA is the entity responsible for key generation, and key distribution. The TA also implements the KP-ABE and PRE modules. It also implements a database for policies and a database for attributes version list and PRE keys.

1. KP-ABE key generation and transfer

This step involves key generation and distribution to the IoT data consumers through HTTPS. The TA is responsible for performing the *Key generation* primitive of the KP-ABE scheme. When a new consumer joins, the TA assigns an access structure to the consumer's key according to an access policy (γ) which is based on some factors like the date of joining the system, the consumer type (i.e., landlord, tenant, or guest), and the type of the data he can access, then he constructs the policy based on the *Attributes* universe (U) stored in *Attributes* storage unit which are also defined by the TA. The consumer then uses the sent decryption key to access the data. The TA stores the policy in *Policies* database to be accessed later in the case of policy updates. The *Attribute Version List* is initiated with the first version of attributes and sent to NOS.

2. KP-ABE proxy re-encryption

This step involves key revocation and re-encryption of old ciphertexts to prevent their compromise. The TA determines a minimal set of attributes stored in AVL without which the access structure of the revoked key can never be satisfied. Then it calls the *UpdateAttribute* primitive, then the master key (MK) and public key (EK) components are redefined accordingly and the PRE key is sent to NOS, then NOS performs the re-encryption. Then, the corresponding decryption key components will be updated by calling *UpdateDK* function and the decryption key will be sent to the affected consumers. The process is shown in figure 4-4.

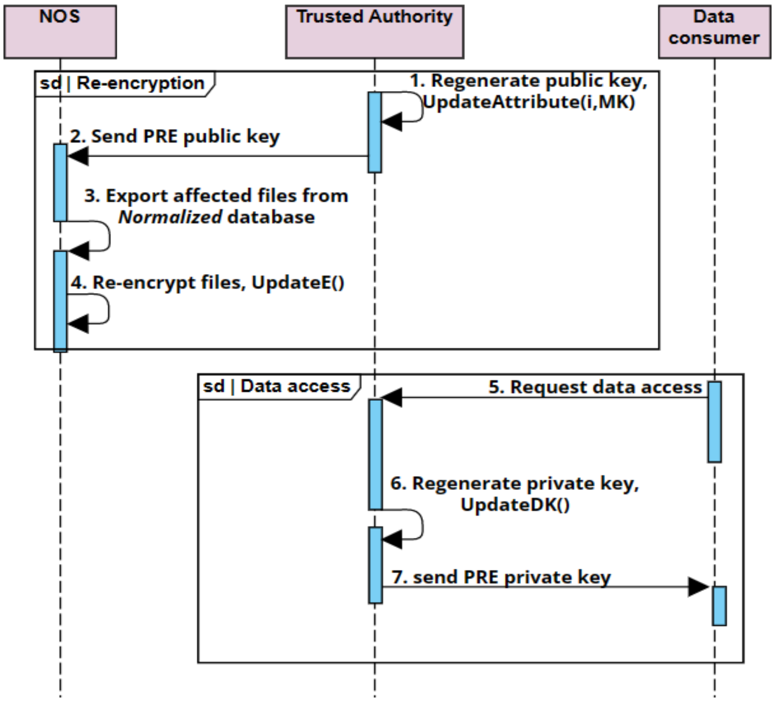

NOS
Trusted Authority
Data consumer
sd | Re-encryption
1. Regenerate public key, UpdateAttribute(i,MK)
2. Send PRE public key
3. Export affected files from *Normalized* database
4. Re-encrypt files, UpdateE()
sd | Data access
5. Request data access
6. Regenerate private key, UpdateDK()
7. send PRE private key

Figure 4-4: Sequence diagram of proxy re-encryption

4.3.3 The data consumers

KP-ABE and PRE decryption

When a new consumer joins the system, he requests decryption keys and access to data from the TA. The TA then sends the decryption keys and PRE keys. Then the user requests access and NOS exports the data from the normalized data database and the user decrypts data using the PRE key first, then the KP-ABE decryption key. The process is shown in figure 4-5.

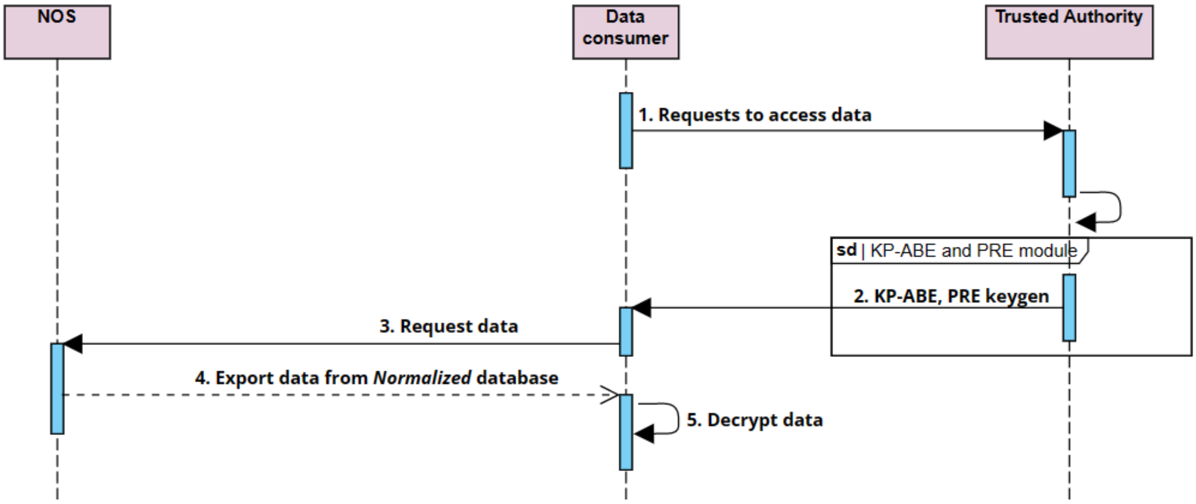

NOS
Data consumer
Trusted Authority
1. Requests to access data
sd | KP-ABE and PRE module
2. KP-ABE, PRE keygen
3. Request data
4. Export data from *Normalized* database
5. Decrypt data

Figure 4-5: Sequence diagram of data retrieval

The proposed model artefact representing the IoT security architecture is shown in figure 4-6.

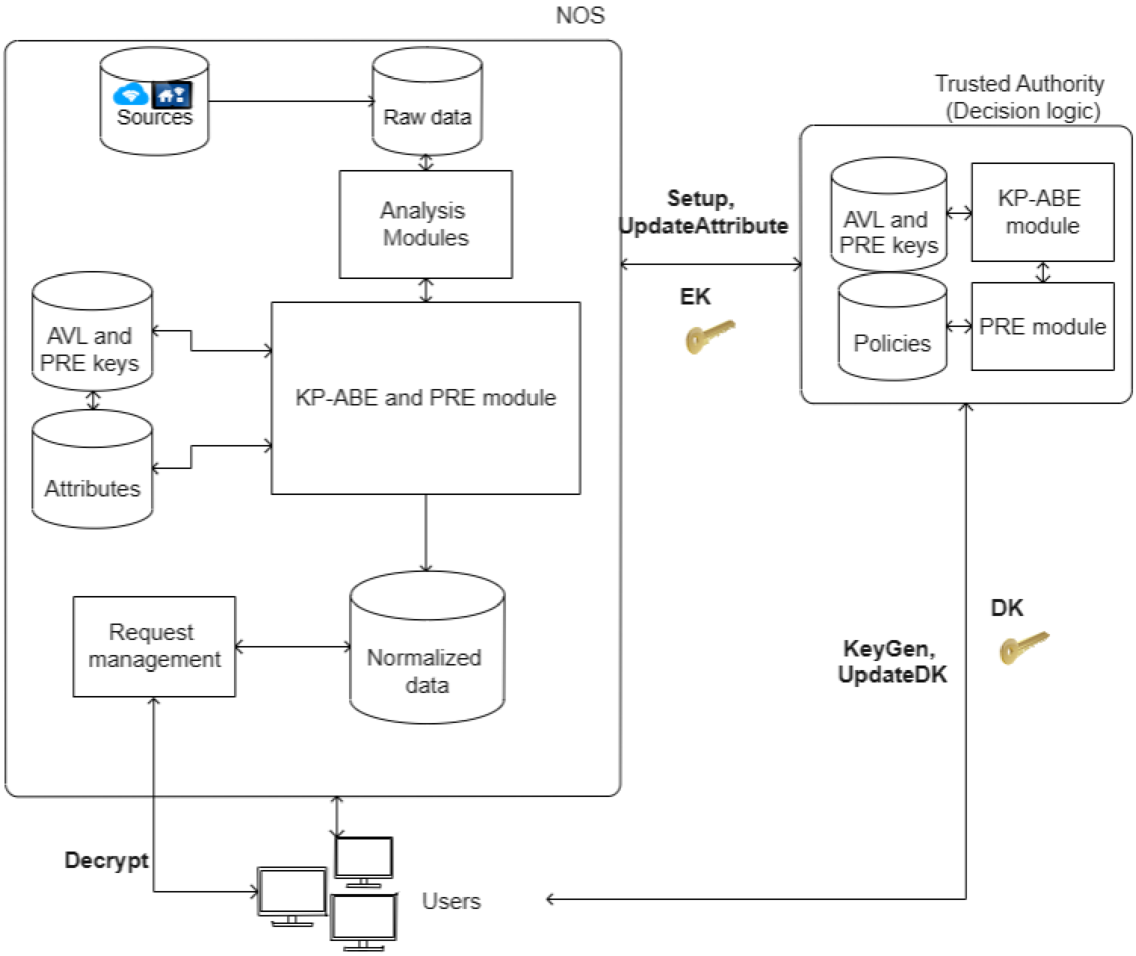


Figure 4-6: Proposed IoT security architecture

The thesis proposed artefact is a model that can be used to produce an instantiation and used in practice in IoT environments. The IoT environment contain the same components proposed by the model except that data will be generated in real-time, consumers will request data using a web application that uses a request management module, and each entity will be implemented in a separate device. To instantiate the model and use it in an IoT environment, the development steps proposed by this thesis can be followed and the instantiation can be used in practice.

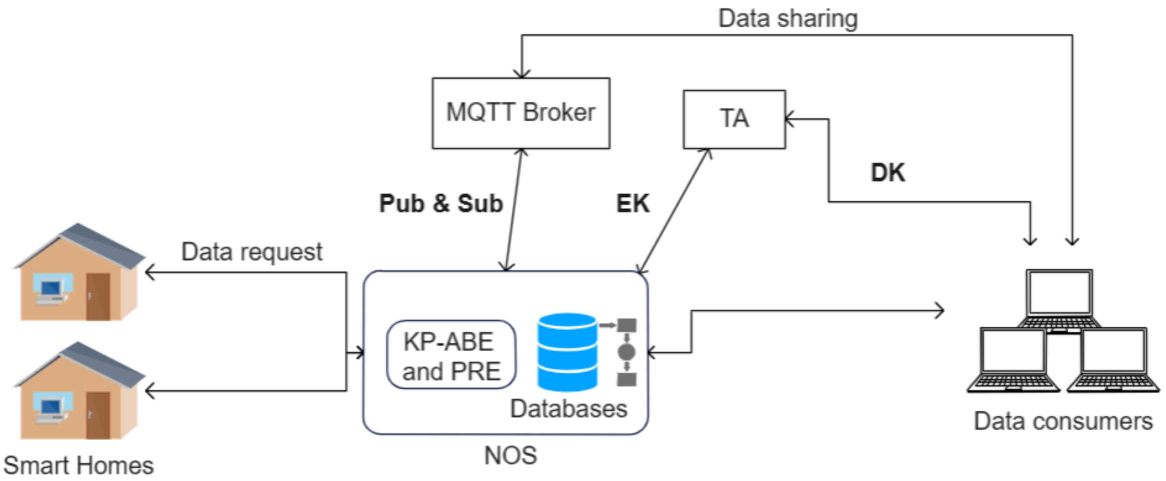


Figure 4-7: IoT Environment example

Figure 4-7 shows an example of an IoT environment instantiating the proposed artefact. NOS middleware fetches the data from the IoT nodes of the smart homes, after that data is analyzed, encrypted, and stored in NOS. Data consumers joining the system request decryption keys from the TA then send access requests to NOS through a request management module that can be implemented in a web application. A request management module can use MQTT protocol which is based on publish-subscribe mechanism to disseminate data to users.

4.4 Demonstrate Artefact

The author used a dataset for demonstrating the artefact. The dataset contains data from a real smart home testbed, such data regard some smart meters installed in three smart homes like electricity and temperature meters. The components of the model were demonstrated on a laptop which installed two operating systems: (i) Ubuntu 22.04.2 LTS (emulating the TA and the users); (ii) Ubuntu 22.04.2 LTS (emulating a Raspberry Pi that installs NOS). The laptop used Wi-Fi network for deploying the local host connection used by the different entities. The KP-ABE and PRE primitives were installed along with NOS on the Ubuntu OS. The NOS modules of the IoT model construction were implemented by means of Node.JS platform for web requests, MongoDB for database storage, Python for automating shell scripts, and GCC-GNU compiler for executing KP-ABE and PRE primitives. The data producers were represented by the dataset which was downloaded from an IoT smart home website[1].

The proposed IoT security model is composed of three modules: the TA, NOS, and the users. NOS is responsible for data encryption and storage and the TA is responsible for key generation while the users request data access, retrieve decryption keys, then decrypt data. Thus, to demonstrate the model design components, the author demonstrated three processes: (1) NOS analysis, encryption, and storage, (2) Data retrieval, and (3) Proxy Re-encryption. According to Johannessen and Perjons (2014), to demonstrate the viability of the artefact the author can choose a real-life case to apply the artefact and document the outcome of it. The demonstration activity is summarized in figure 4-8.

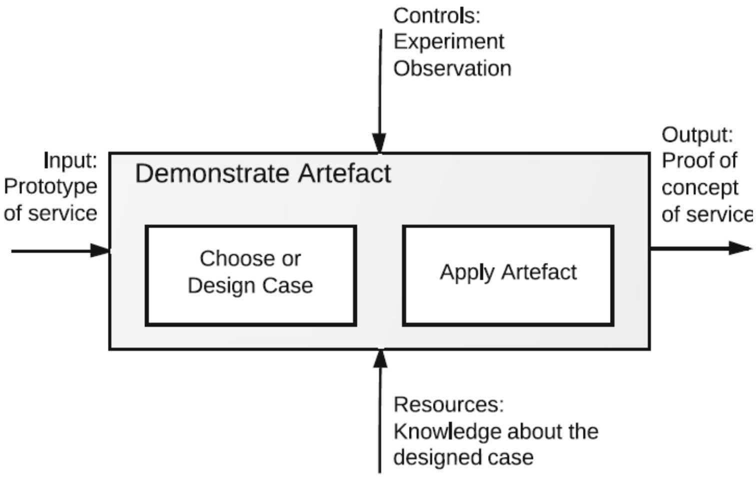


Figure 4-8: Demonstrate artefact activities from Johannessen and Perjons (2014)

[1] Smart - UMass Trace Repository

4.4.1 Design Case

The author used datasets from three smart homes (i.e., Home A, Home B, Home C), the dataset of each home included electrical data (i.e., kitchen, lights, heater, washing machine, etc..) and weather data (i.e., temperature, humidity, visibility, pressure, etc..). The author associated each dataset with several attributes which encrypt the dataset along with the encryption key (EK). The attributes of the datasets obtained from the data producers are described as follows:

- Electrical data (*EleX*), which implies that the dataset contains electrical data gathered from house X nodes. X can represent houses A, B or C.
- Weather data (*WeaX*), which implies that the dataset contains weather data gathered from house X nodes, X can represent houses A, B or C.
- Access type (*AccX*), which implies that the dataset can be accessed by a consumer X, X can represent consumer type: L for landlord, T for tenant, or G for guest.
- Starting date of data (*SDatY*), which represents the starting date of collecting the data and it has a numerical value which represents the year number.
- Starting date of data (*SDatM*), which represents the starting date of collecting the data and it has a numerical value which represents the month number.
- End date of data (*EDatY*), which represents the end date of collecting the data and it has a numerical value which represents the year number.
- End date of data (*EDatM*), which represents the end date of collecting the data and it has a numerical value which represents the month number.

Electricity dataset contains data about the energy consumption of all the electrical devices inside the house. Only the landlord and the tenant can access these data. An example of an attribute set γ for the electricity dataset of Home A gathered in January in 2016 is the following:

γ (Home A-electricity-January) = {*EleA*, *AccL*, *SDatM* = 1, *SDatY* = 16, *EDatM* = 2, *EDatY* = 16)}

To access these data, the TA will provide the consumer with an access policy τ embedded in the private decryption key (DK). For a new tenant who joined home A on February and will leave on May and wants to access the electrical data, the TA will embed the following policy to the decryption key:

τ (Tenant-February) = {*EleA* and *AccT* and (*SDatM* >1 and *EDatM* < 5)}

The author derived the policies from the attributes defined above and used them to demonstrate and evaluate the artifact.

4.4.2 Apply Artifact

4.4.2.1 NOS Middleware

Data security and quality

In the first step, the security model aims to improve data quality, and security by filtering data which originated from a credible and trustworthy source. Here, NOS which is emulated on Ubuntu as a Raspberry Pi fetches the data from the data sources, checks for its quality and security, and assigns scores for security (i.e., authentication, privacy, confidentiality, integrity), and quality (i.e., accuracy, freshness, completeness).

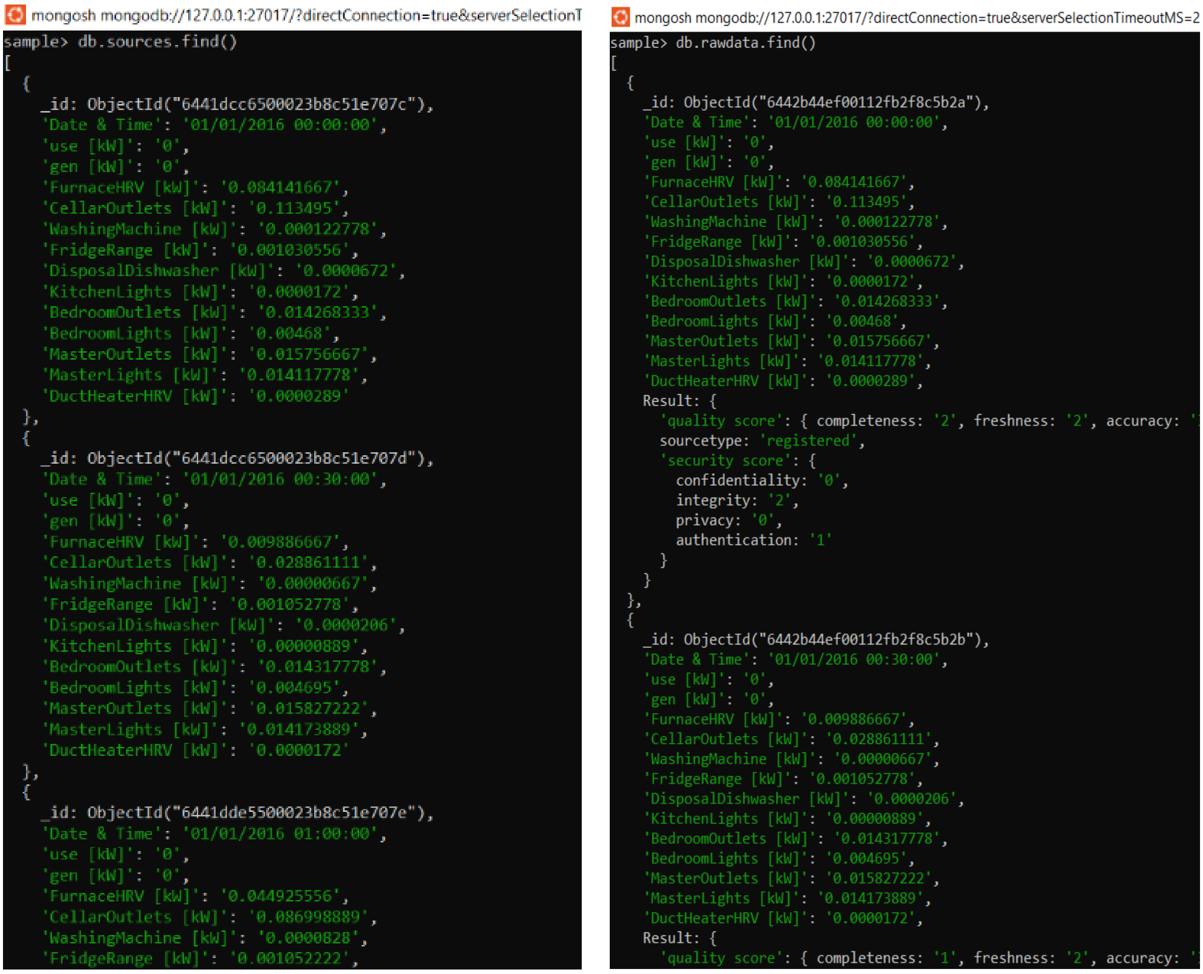
Data sources data are stored in a data storage unit known as *Sources* collection. For demonstration purposes, the author filled *Sources* database with the electricity dataset of smart home A gathered in January 2016 and added some sources information which will be used to filter data of most quality. For example, one data source can have the following data associated with a dataset:

{"id": 1, "type": "WSN", "communication": "WIFI", "algorithm": {“type": "ECC", "score": 10}}

The steps of data analysis are as follows:

- Source Analysis
    - First, the data source is checked, if it is a registered one then authentication of the source is required, and the data is decrypted.
    - A score is assigned based on the encryption scheme used (ECC, RSA, PKI).
- Security Analysis
    - Confidentiality and privacy are evaluated according to the encryption schema used.
    - Integrity is checked and a score is assigned (0 for violated, 2 for unviolated).
- Quality Analysis
    - Timeliness score is set, which is the time between when the data was sampled to when it’s sent to NOS.
    - Completeness score is set, which is the ratio between number of collected values and expected values.
    - Accuracy value is set, which is based on the difference between the sampled value and a reference value.
    - Precision is set, which is characterized in terms of the standard deviation of the sampled values.

Raw data database entries are shown in figure 4-9 before and after analyzing data.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionT
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2

```
sample> db.sources.find()
[
  {
    _id: ObjectId("6441dcc6500023b8c51e707c"),
    'Date & Time': '01/01/2016 00:00:00',
    'use [kW]': '0',
    'gen [kW]': '0',
    'FurnaceHRV [kW]': '0.084141667',
    'CellarOutlets [kW]': '0.113495',
    'WashingMachine [kW]': '0.000122778',
    'FridgeRange [kW]': '0.001030556',
    'DisposalDishwasher [kW]': '0.0000672',
    'KitchenLights [kW]': '0.0000172',
    'BedroomOutlets [kW]': '0.014268333',
    'BedroomLights [kW]': '0.00468',
    'MasterOutlets [kW]': '0.015756667',
    'MasterLights [kW]': '0.014117778',
    'DuctHeaterHRV [kW]': '0.0000289'
  },
  {
    _id: ObjectId("6441dcc6500023b8c51e707d"),
    'Date & Time': '01/01/2016 00:30:00',
    'use [kW]': '0',
    'gen [kW]': '0',
    'FurnaceHRV [kW]': '0.009886667',
    'CellarOutlets [kW]': '0.028861111',
    'WashingMachine [kW]': '0.00000667',
    'FridgeRange [kW]': '0.001052778',
    'DisposalDishwasher [kW]': '0.0000206',
    'KitchenLights [kW]': '0.00000889',
    'BedroomOutlets [kW]': '0.014317778',
    'BedroomLights [kW]': '0.004695',
    'MasterOutlets [kW]': '0.015827222',
    'MasterLights [kW]': '0.014173889',
    'DuctHeaterHRV [kW]': '0.0000172'
  },
  {
    _id: ObjectId("6441dde5500023b8c51e707e"),
    'Date & Time': '01/01/2016 01:00:00',
    'use [kW]': '0',
    'gen [kW]': '0',
    'FurnaceHRV [kW]': '0.044925556',
    'CellarOutlets [kW]': '0.086998889',
    'WashingMachine [kW]': '0.0000828',
    'FridgeRange [kW]': '0.001052222',
```

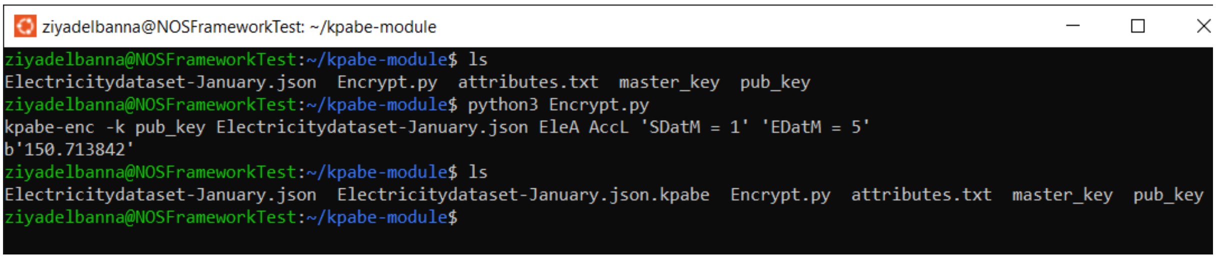
```
sample> db.rawdata.find()
[
  {
    _id: ObjectId("6442b44ef00112fb2f8c5b2a"),
    'Date & Time': '01/01/2016 00:00:00',
    'use [kW]': '0',
    'gen [kW]': '0',
    'FurnaceHRV [kW]': '0.084141667',
    'CellarOutlets [kW]': '0.113495',
    'WashingMachine [kW]': '0.000122778',
    'FridgeRange [kW]': '0.001030556',
    'DisposalDishwasher [kW]': '0.0000672',
    'KitchenLights [kW]': '0.0000172',
    'BedroomOutlets [kW]': '0.014268333',
    'BedroomLights [kW]': '0.00468',
    'MasterOutlets [kW]': '0.015756667',
    'MasterLights [kW]': '0.014117778',
    'DuctHeaterHRV [kW]': '0.0000289',
    Result: {
      'quality score': { completeness: '2', freshness: '2', accuracy: '
      sourcetype: 'registered',
      'security score': {
        confidentiality: '0',
        integrity: '2',
        privacy: '0',
        authentication: '1'
      }
    }
  },
  {
    _id: ObjectId("6442b44ef00112fb2f8c5b2b"),
    'Date & Time': '01/01/2016 00:30:00',
    'use [kW]': '0',
    'gen [kW]': '0',
    'FurnaceHRV [kW]': '0.009886667',
    'CellarOutlets [kW]': '0.028861111',
    'WashingMachine [kW]': '0.00000667',
    'FridgeRange [kW]': '0.001052778',
    'DisposalDishwasher [kW]': '0.0000206',
    'KitchenLights [kW]': '0.00000889',
    'BedroomOutlets [kW]': '0.014317778',
    'BedroomLights [kW]': '0.004695',
    'MasterOutlets [kW]': '0.015827222',
    'MasterLights [kW]': '0.014173889',
    'DuctHeaterHRV [kW]': '0.0000172',
    Result: {
      'quality score': { completeness: '1', freshness: '2', accuracy: '
```

Figure 4-9: Before and after analyzing data

KP-ABE

For the second step, the model has an encryption module that applies KP-ABE. Following the previous step, data are exported in JSON format because it is compatible with MongoDB and is a lightweight data-interchange format and is both easy for humans to read and for machines to parse and generate. The KP-ABE module gets data from local host storage and uses the attribute set defined in the case design section 4.4.1. The TA executes **Setup** primitive to produce the encryption key and send it over HTTPS to NOS. NOS executes KP-ABE operations using Python and saves the encrypted data along with the associated attributes in *Normalized data* database.

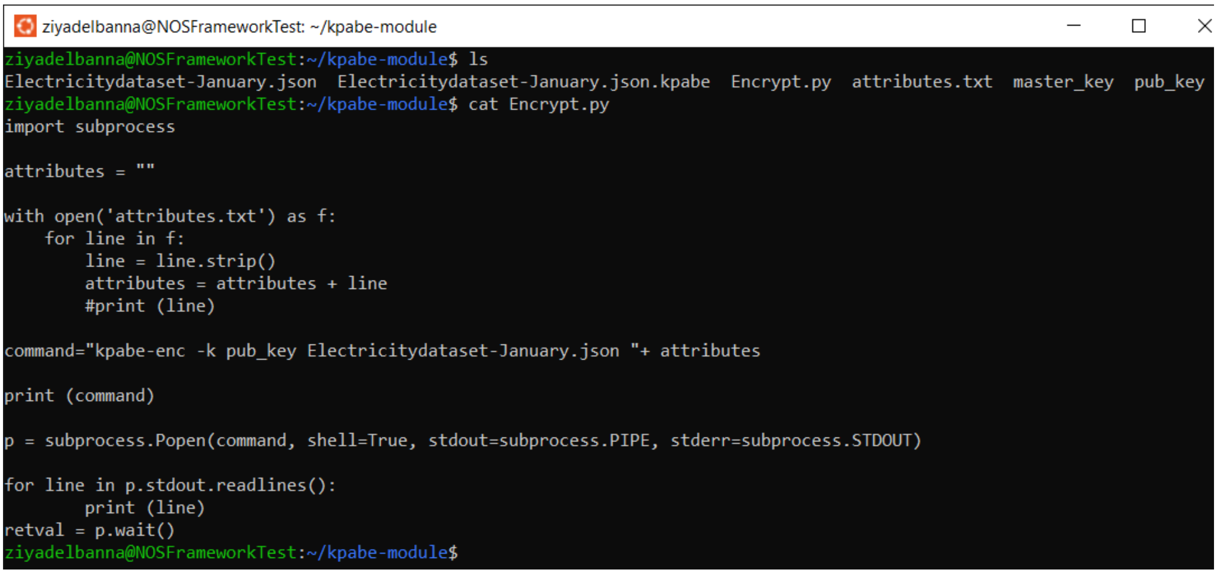
Figure 4-10 shows how the **Encrypt** primitive is executed by means of Python. For executing the encryption primitive, the required attributes are fetched from *Attributes* database. NOS uses a python automation script to read and encrypt data according to the defined attributes.



```
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$ ls
Electricitydataset-January.json  Encrypt.py  attributes.txt  master_key  pub_key
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$ python3 Encrypt.py
kpabe-enc -k pub_key Electricitydataset-January.json EleA AccL 'SDatM = 1' 'EDatM = 5'
b'150.713842'
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$ ls
Electricitydataset-January.json  Electricitydataset-January.json.kpabe  Encrypt.py  attributes.txt  master_key  pub_key
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$
```

Figure 4-10: KP-ABE encryption and attributes retrieval

The python script used for retrieving the attributes from the file in *Attributes* database in an automated manner is shown in figure 4-11.



```
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$ ls
Electricitydataset-January.json  Electricitydataset-January.json.kpabe  Encrypt.py  attributes.txt  master_key  pub_key
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$ cat Encrypt.py
import subprocess

attributes = ""

with open('attributes.txt') as f:
    for line in f:
        line = line.strip()
        attributes = attributes + line
        #print (line)

command="kpabe-enc -k pub_key Electricitydataset-January.json "+ attributes

print (command)

p = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)

for line in p.stdout.readlines():
        print (line)
retval = p.wait()
ziyadelbanna@NOSFrameworkTest:~/kpabe-module$
```

Figure 4-11: KP-ABE encryption and attributes retrieval-2

After encrypting the dataset, KP-ABE module stores each encrypted dataset (i.e., Electricity-January, Electricity-March, Weather-February, etc...) in *Normalized* database.

4.4.2.2 The TA

Proxy Re-encryption

This step describes how the new proposed model performs PRE. PRE module was introduced in this model as a re-encryption method due to its efficient operations compared to the naïve re-encryption methods. Moreover, one of the research gaps of the previous model proposed by Sicari et al. (2020) is the need to provide an efficient, and secure re-encryption method to avoid performing all ABE operations (i.e., key setup, key generation, and key encryption) again every time a key is compromised, or a user is revoked. To achieve the research purposes, a PRE module based on Yu et al.'s (2010) scheme is introduced in the model.

The process starts with the TA whose behaviour is emulated by means of Ubuntu platform connected to the same network as the NOS middleware, the TA first generates the *master* (master_key) and *public* keys (pub_key) by executing the **Setup** primitive. The input to the setup primitive is the *Attributes* universe defined and decided by the TA. The TA then generates a private key (priv_key) for the consumer using the defined access structure policy.

In case of a key compromise the TA generates a re-encryption key using **UpdateAttribute** primitive and sends it to the NOS middleware to re-encrypt old ciphertexts using **UpdateE** and stores the key in *PRE keys* database and the new attributes in the *AVL*. The TA also updates the decryption key using **UpdateDK** primitive and sends it to the affected consumers. The steps are shown in figures 4-12 and 4-13 below.

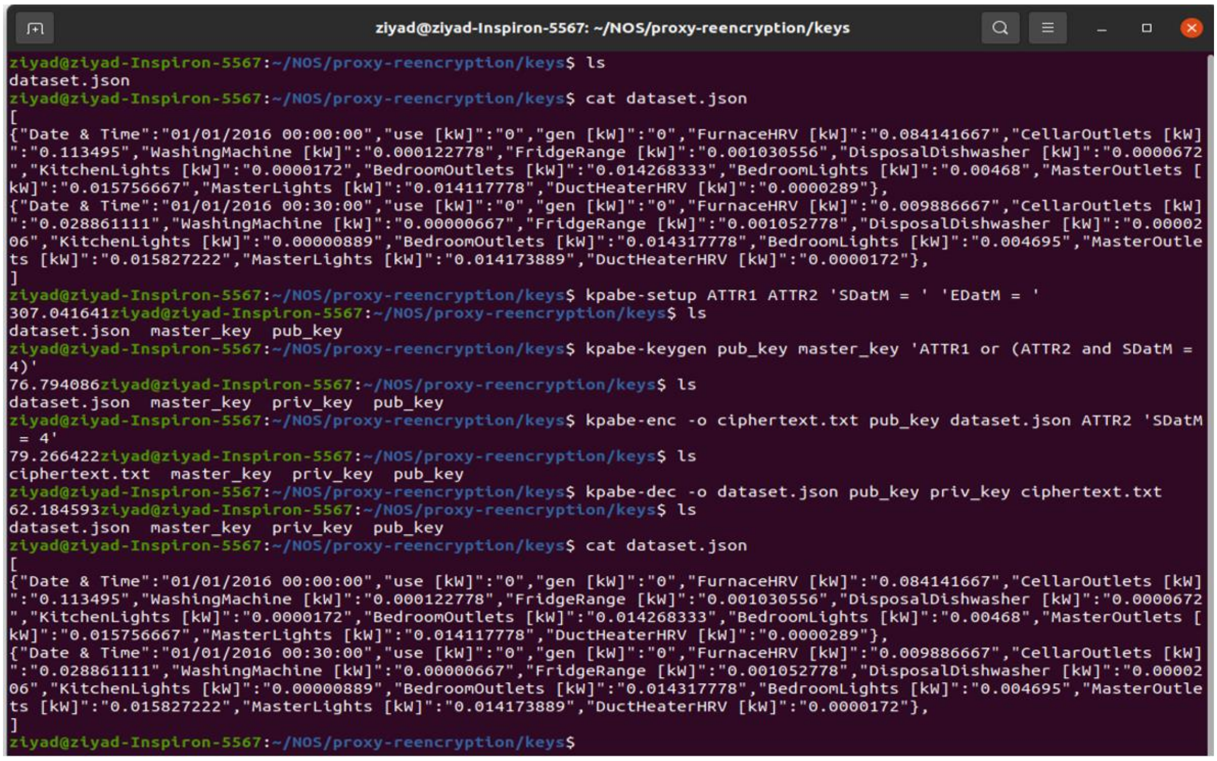
The initial steps of the PRE scheme are as follows:

- First, the TA generates a PRE public key, encryption key, and master key.

- Second, the TA sends the attributes, and the PRE public key along with the encryption key to the NOS middleware.
- Third, the NOS encrypts the dataset using the encryption key and the attributes and exports the encrypted dataset as a ciphertext stored in a file.
- Fourth, the resulting ciphertext is encrypted using the latest version of the PRE key sent by the TA and stored in the *Normalized data* database.

For decryption, the steps are as follows (Consumers):

- First, the TA generates decryption key, PRE private key and sends them to the consumer.
- Second, the user decrypts the ciphertexts and the KP-ABE encrypted file is generated.
    - If the user can't decrypt using the PRE key, then either the user is revoked, or the access request is malicious due to a key compromise.
- The user then can have access to the data according to the policy defined by the TA by decrypting the generated KP-ABE encrypted file.

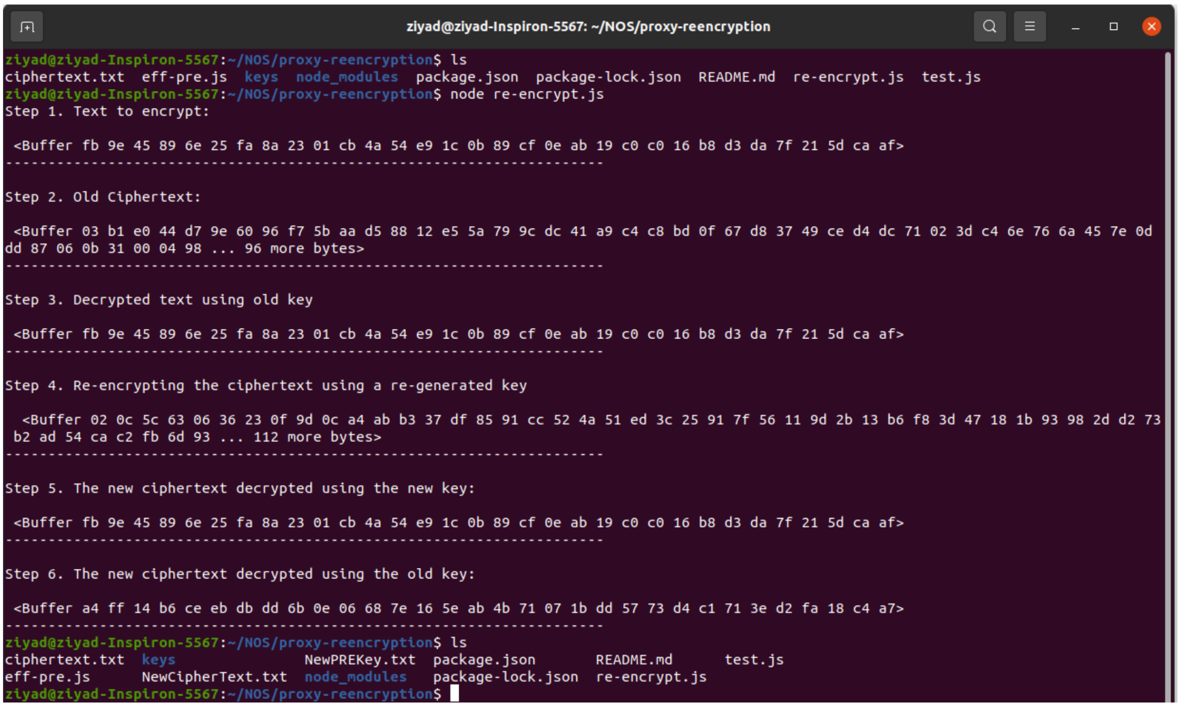


ziyad@ziyad-Inspiron-5567: ~/NOS/proxy-reencryption/keys

```
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ ls
dataset.json
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ cat dataset.json
[
{"Date & Time":"01/01/2016 00:00:00","use [kW]":"0","gen [kW]":"0","FurnaceHRV [kW]":"0.084141667","CellarOutlets [kW]
":"0.113495","WashingMachine [kW]":"0.000122778","FridgeRange [kW]":"0.001030556","DisposalDishwasher [kW]":"0.0000672
","KitchenLights [kW]":"0.0000172","BedroomOutlets [kW]":"0.014268333","BedroomLights [kW]":"0.00468","MasterOutlets [
kW]":"0.015756667","MasterLights [kW]":"0.014117778","DuctHeaterHRV [kW]":"0.0000289"},
{"Date & Time":"01/01/2016 00:30:00","use [kW]":"0","gen [kW]":"0","FurnaceHRV [kW]":"0.009886667","CellarOutlets [kW]
":"0.028861111","WashingMachine [kW]":"0.00000667","FridgeRange [kW]":"0.001052778","DisposalDishwasher [kW]":"0.00002
06","KitchenLights [kW]":"0.00000889","BedroomOutlets [kW]":"0.014317778","BedroomLights [kW]":"0.004695","MasterOutle
ts [kW]":"0.015827222","MasterLights [kW]":"0.014173889","DuctHeaterHRV [kW]":"0.0000172"},
]
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ kpabe-setup ATTR1 ATTR2 'SDatM = ' 'EDatM = '
307.041641ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ ls
dataset.json  master_key  pub_key
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ kpabe-keygen pub_key master_key 'ATTR1 or (ATTR2 and SDatM =
4)'
76.794086ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ ls
dataset.json  master_key  priv_key  pub_key
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ kpabe-enc -o ciphertext.txt pub_key dataset.json ATTR2 'SDatM
 = 4'
79.266422ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ ls
ciphertext.txt  master_key  priv_key  pub_key
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ kpabe-dec -o dataset.json pub_key priv_key ciphertext.txt
62.184593ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ ls
dataset.json  master_key  priv_key  pub_key
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$ cat dataset.json
[
{"Date & Time":"01/01/2016 00:00:00","use [kW]":"0","gen [kW]":"0","FurnaceHRV [kW]":"0.084141667","CellarOutlets [kW]
":"0.113495","WashingMachine [kW]":"0.000122778","FridgeRange [kW]":"0.001030556","DisposalDishwasher [kW]":"0.0000672
","KitchenLights [kW]":"0.0000172","BedroomOutlets [kW]":"0.014268333","BedroomLights [kW]":"0.00468","MasterOutlets [
kW]":"0.015756667","MasterLights [kW]":"0.014117778","DuctHeaterHRV [kW]":"0.0000289"},
{"Date & Time":"01/01/2016 00:30:00","use [kW]":"0","gen [kW]":"0","FurnaceHRV [kW]":"0.009886667","CellarOutlets [kW]
":"0.028861111","WashingMachine [kW]":"0.00000667","FridgeRange [kW]":"0.001052778","DisposalDishwasher [kW]":"0.00002
06","KitchenLights [kW]":"0.00000889","BedroomOutlets [kW]":"0.014317778","BedroomLights [kW]":"0.004695","MasterOutle
ts [kW]":"0.015827222","MasterLights [kW]":"0.014173889","DuctHeaterHRV [kW]":"0.0000172"},
]
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption/keys$
```

Figure 4-12: KPABE setup, key generation, encryption, and decryption executed by the TA

Now suppose that PRE key is compromised for any reason, in this case to prevent unauthorized access, the TA will generate a re-encryption key, and send it to NOS to re-encrypt **ciphertext.txt** as shown in figure 4-13. First, the ciphertext in ciphertext.txt is re-encrypted using a new key generated by the TA. The first step in figure 4-13 shows the old ciphertext to be encrypted (in bytes) and the second step shows the encrypted ciphertext, then from the fourth step a new ciphertext is generated which is not the same as the ciphertext in step 2. This is done after executing **UpdateE** primitive and generating new components to the pre-existing ciphertext. Steps 5 and 6 show that the old key cannot re-decrypt the ciphertext anymore because it has changed. These steps are applied iteratively in the model each time a key is compromised.



```
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption$ ls
ciphertext.txt  eff-pre.js  keys  node_modules  package.json  package-lock.json  README.md  re-encrypt.js  test.js
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption$ node re-encrypt.js
Step 1. Text to encrypt:

 <Buffer fb 9e 45 89 6e 25 fa 8a 23 01 cb 4a 54 e9 1c 0b 89 cf 0e ab 19 c0 c0 16 b8 d3 da 7f 21 5d ca af>
----------------------------------------------------------------------

Step 2. Old Ciphertext:

 <Buffer 03 b1 e0 44 d7 9e 60 96 f7 5b aa d5 88 12 e5 5a 79 9c dc 41 a9 c4 c8 bd 0f 67 d8 37 49 ce d4 dc 71 02 3d c4 6e 76 6a 45 7e 0d
dd 87 06 0b 31 00 04 98 ... 96 more bytes>
----------------------------------------------------------------------

Step 3. Decrypted text using old key

 <Buffer fb 9e 45 89 6e 25 fa 8a 23 01 cb 4a 54 e9 1c 0b 89 cf 0e ab 19 c0 c0 16 b8 d3 da 7f 21 5d ca af>
----------------------------------------------------------------------

Step 4. Re-encrypting the ciphertext using a re-generated key

  <Buffer 02 0c 5c 63 06 36 23 0f 9d 0c a4 ab b3 37 df 85 91 cc 52 4a 51 ed 3c 25 91 7f 56 11 9d 2b 13 b6 f8 3d 47 18 1b 93 98 2d d2 73
 b2 ad 54 ca c2 fb 6d 93 ... 112 more bytes>
----------------------------------------------------------------------

Step 5. The new ciphertext decrypted using the new key:

 <Buffer fb 9e 45 89 6e 25 fa 8a 23 01 cb 4a 54 e9 1c 0b 89 cf 0e ab 19 c0 c0 16 b8 d3 da 7f 21 5d ca af>
----------------------------------------------------------------------

Step 6. The new ciphertext decrypted using the old key:

 <Buffer a4 ff 14 b6 ce eb db dd 6b 0e 06 68 7e 16 5e ab 4b 71 07 1b dd 57 73 d4 c1 71 3e d2 fa 18 c4 a7>
----------------------------------------------------------------------
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption$ ls
ciphertext.txt  keys               NewPREKey.txt  package.json       README.md      test.js
eff-pre.js      NewCipherText.txt  node_modules   package-lock.json  re-encrypt.js
ziyad@ziyad-Inspiron-5567:~/NOS/proxy-reencryption$ 
```

Figure 4-13: PRE of ciphertext

4.5 Evaluate Artefact

The principle aim of the evaluation phase in DSR is to determine how well the result artefact works, and not to theorize or prove why the artefact works. The method used during the evaluation of the designed artefact should focus on why an artefact is successful or not (Hevner et al., 2004). According to Johannessen and Perjons (2014), the evaluation method should determine how well the artefact is able to solve the explicated problem and to what extent it fulfills the requirements and the answer to this question will consist of descriptive knowledge as well as explanatory knowledge. The author followed the DSR evaluation steps presented in figure 4-14.

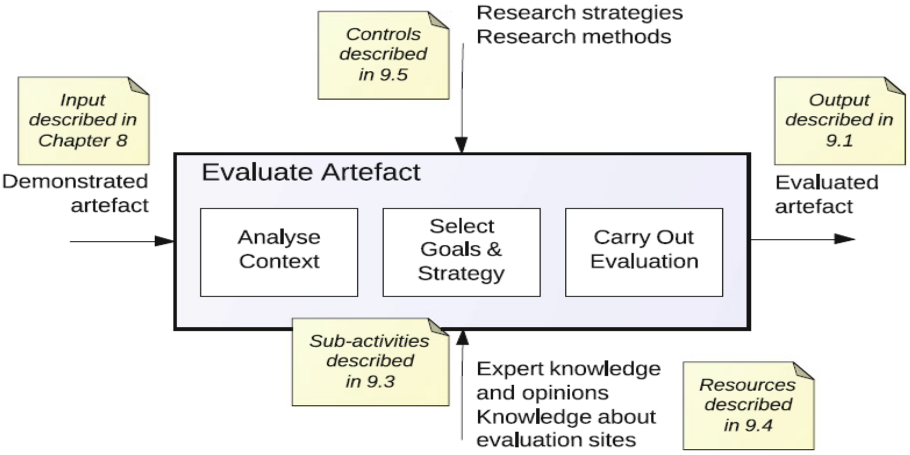


Figure 4-14: Evaluate artefact from Johannesen and Perjons (2014)

To analyze the context and select goals, the author aimed to understand the general quality of the proposed artefact and tackle the research challenges taken from previous work (Sicari et al., 2020).

The research evaluation goal was to analyze two functional requirements which are: (1) Whether KP-ABE enhanced the efficiency of the framework during data-retrieval compared to the previous middleware, (2) The effect of PRE on security and efficiency compared to the previous middleware. Then the author compared the evaluated artefact with the previous existing model. Specifically, the most suitable evaluation method to carry out is an ex-ante evaluation approach with an artificial strategy. The reason for choosing an ex-ante approach is that it is the most suitable evaluation strategy which can be used to evaluate a design or a prototype (Johannessen and Perjons, 2014). The artificial evaluation carried out was a computer simulation employing ABE and PRE using parameters of increased complexity to assess the effectiveness of the proposed model.

Evaluation metrics

To address **RQ1**, the author enacted a controlled experiment by a computer simulation using the same datasets used in the previous case study.

The first metric to evaluate to address **RQ1** was the data retrieval delay. The data retrieval delay means the time from when a consumer joins the IoT system to when he decrypts the data. The steps of retrieving the data are as follows:

- First, the data consumer requests a decryption key from the TA, who defines an access policy (γ) which is based on some factors like the date of joining the system, the consumer type (i.e., landlord, tenant, or guest), and the type of the data he can access.
- The TA then executes the KP-ABE primitive **KeyGen** generating a decryption key based on γ.
- The consumer then requests the data from NOS which fetches the encrypted data and provides it to the user who decrypts it.

The author evaluated the performance of the proposed model by varying the number of attributes used in the KP-ABE primitives introduced in the model and compared it to the performance of the CP-ABE primitives used by the previous model. The maximum number of attributes used in the smart home application is about 50 attributes, thus, the author limited the evaluation to a maximum of 50 attributes and varied the number of attributes to reflect how increasing the number will affect the ABE operations. Measurements were taken five times for 10, 20, 30, 40 and 50 attributes, then an average was calculated, and results were plotted on a line graph.

Execution time

The execution time of cryptographic operations is the most important metric required to evaluate the performance of ABE. The author tested the execution time of the three major cryptographic operations of the KP-ABE scheme used in the model vs the CP-ABE scheme used in the previous model.

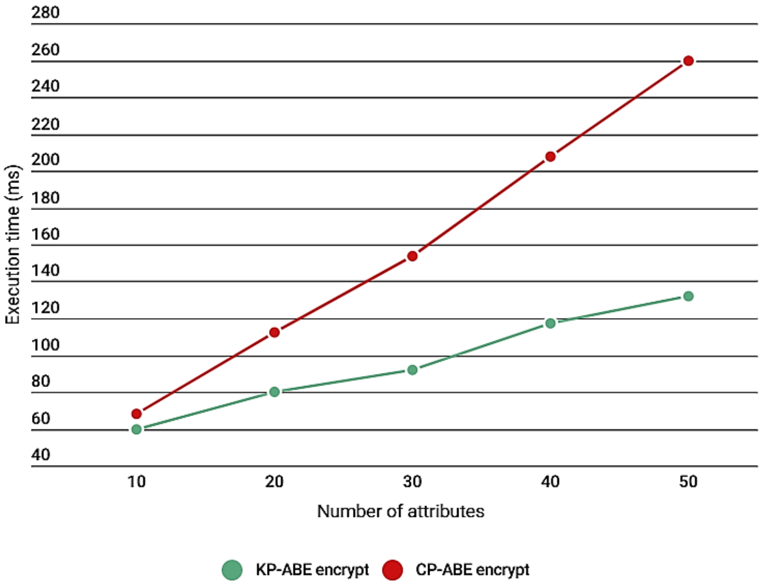


Figure 4-15: ABE-encrypt execution time

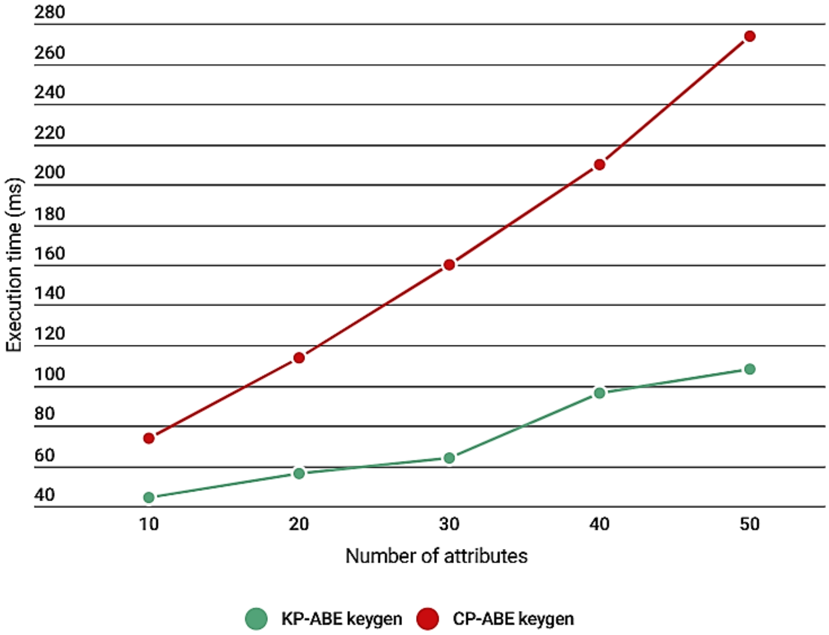


Figure 4-16: ABE-keygen execution time

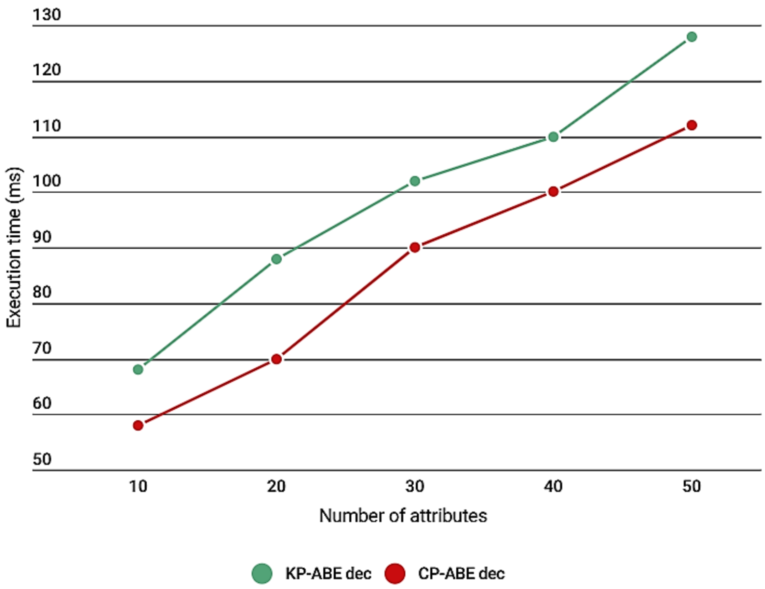


Figure 4-17: ABE-decrypt execution time

Figures 4-15, 16, and 17 show that the execution time of the three operations increase linearly with the number of attributes and that KP-ABE operations are faster than CP-ABE operations except for the decrypt algorithm where they approximately take the same execution time.

Data retrieval delay

The process of data retrieval in the proposed model is the same as the process in the old model. The steps of data retrieval are done each time a new user joins and asks for data access. The author tested the data retrieval process and compared the results with that of the old model.

Figure 4-18 shows that the average retrieval data delay using the proposed model is 717 ms, while that of the old model is 1060 ms. This implies that the proposed scheme in the new model is 32% faster than the old model.

It appears evident that using the KP-ABE scheme constructed by Goyal et. al (2006) in NOS for IoT smart home datasets introduces less delay when consumers are trying to retrieve their data. Thus, the execution time and data retrieval delay provided an answer to **RQ1**, as one can evidently conclude that the proposed model can improve the efficiency of ABE operations in IoT applications.

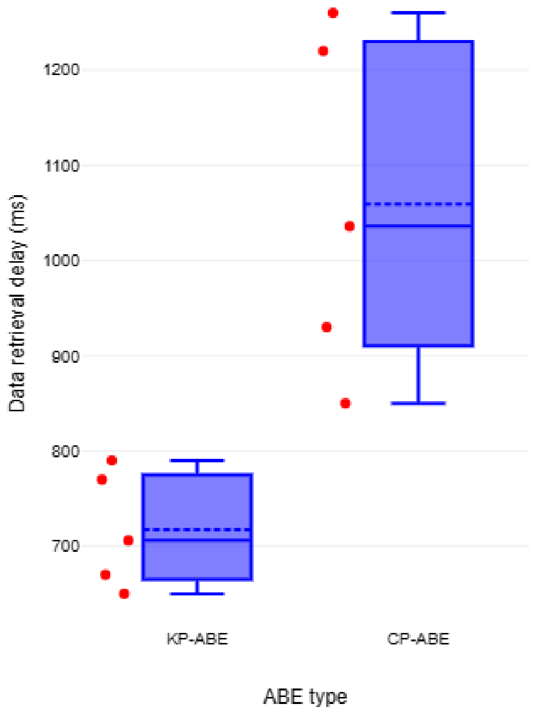


Figure 4-18: Whiskers-box diagram of mean data retrieval delay comparison: KP-ABE vs CP-ABE approach

CPU Load

The second most important metric to evaluate the efficiency of ABE operations in lightweight IoT devices is the CPU load required when performing each of these operations. The author obtained the results programmatically by executing a shell script that executes the operations and measures the required CPU load for each operation by varying the number of attributes.

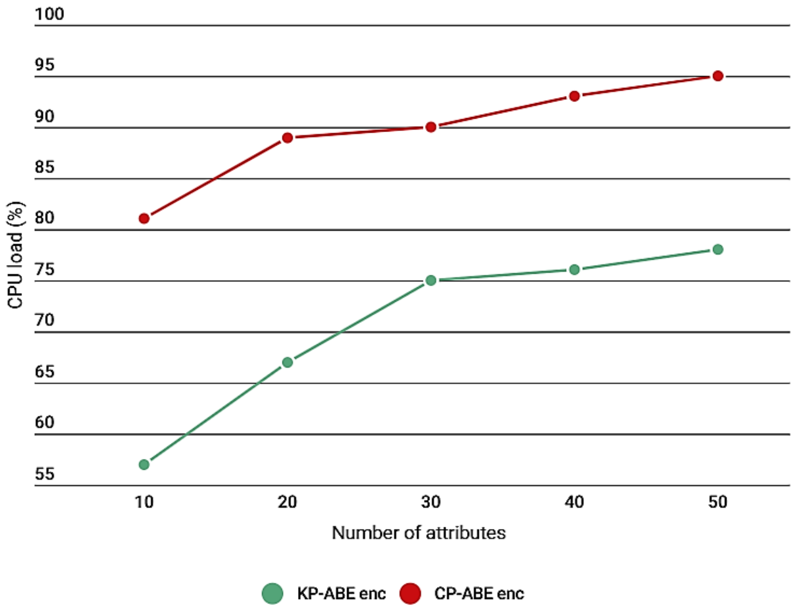


Figure 4-19: CPU load of ABE-encrypt

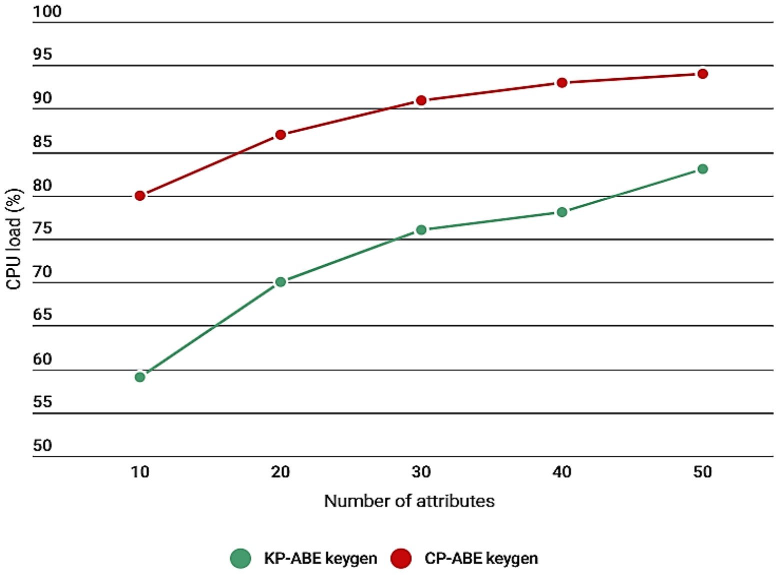


Figure 4-20: CPU load of ABE-keygen

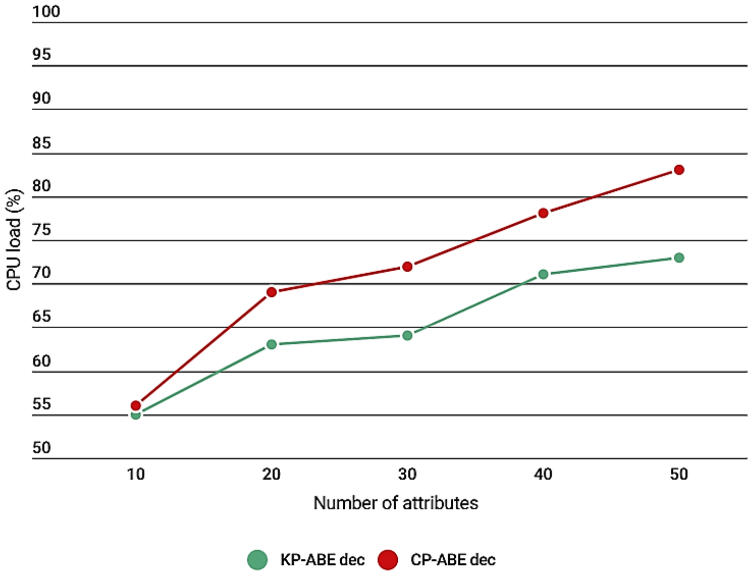


Figure 4-21: CPU load of ABE-decrypt

Figures 4-19, 20, and 21 show that the CPU load of IoT devices required to execute KP-ABE operations is less than that required for CP-ABE operations.

Memory consumption

Another factor affecting the efficiency of ABE operations on IoT devices is the size of the memory or RAM allocated when performing a certain operation. The author tested memory consumption as another evaluation metric for assessing the difference between the two schemes used in the middleware. The summary of the evaluation results is shown in table 4-1.

Operation		Attributes	Memory allocation (MB)
CP-ABE	Encrypt	10	11
		100	11
		1000	17
	Keygen	10	11
		100	11
		1000	15
	Decrypt	10	10
		100	10
		1000	13
KP-ABE	Encrypt	10	11
		100	11
		1000	12
	Keygen	10	11
		100	11
		1000	13
	Decrypt	10	11
		100	11
		1000	12

Table 4-1: Memory consumption of ABE schemes

The table shows that the size of memory/RAM allocated to perform the two ABE types in IoT devices is almost the same, with the average for KP-ABE memory consumption being slightly less than CP-ABE. The average for KP-ABE is (~11MB) and for CP-ABE is (~13MB). Such memory allocation is acceptable considering the size of today's IoT devices.

CPU load and memory consumption evaluation results show that the new model's scheme constructed by Goyal et. al (2006) is more efficient than the previous scheme. Thus, the results provided an answer to **RQ1**.

To address **RQ2**, the author enacted a controlled experiment on a computer simulation and tested the proxy re-encryption process.

Proxy Re-encryption

To evaluate the scheme in terms of efficiency, the old mechanism of key revocation is compared to the PRE scheme used in the proposed model. In the old CP-ABE scheme, when a user is revoked, the TA sends an updated list of the attributes' version list to NOS, NOS then executes CP-ABE encrypt primitive for the datasets using the new attributes and the TA regenerates a decryption key for each affected consumer and sends it to them. For each key revocation, supposing *n* consumers, the TA sends one list update and generates *a*n* keys, where a ∈ [0, 1] is the ratio of affected consumers. Figure 4-21 shows the time of re-encryption and key generation in both schemes for n = 50, a = 0.6, and 50 attributes.

Figure 4-22 shows that for 30 affected consumers, the average time taken to re-generate the encryption keys and re-encrypt the data in the proposed PRE scheme is 1 second while that of the old method is 7.75 seconds, which means that the proposed PRE scheme is 87% faster than the old key revocation method.

In terms of security, the proposed PRE scheme not only generates new keys for encrypting new data, but it also executes **UpdateE** primitive which re-encrypts old ciphertexts in case a user is revoked, or a key is compromised. In this manner, only the newly generated keys will be able to decrypt old data, while any other keys will not be able to decrypt.

The experiment provided an answer to **RQ2**, as one can evidently conclude that the proposed PRE scheme's effect in terms of efficiency is that it is significantly more efficient than the old key revocation method. One can also conclude that using the proposed model will enhance the security in IoT applications in case of key revocation or compromise.

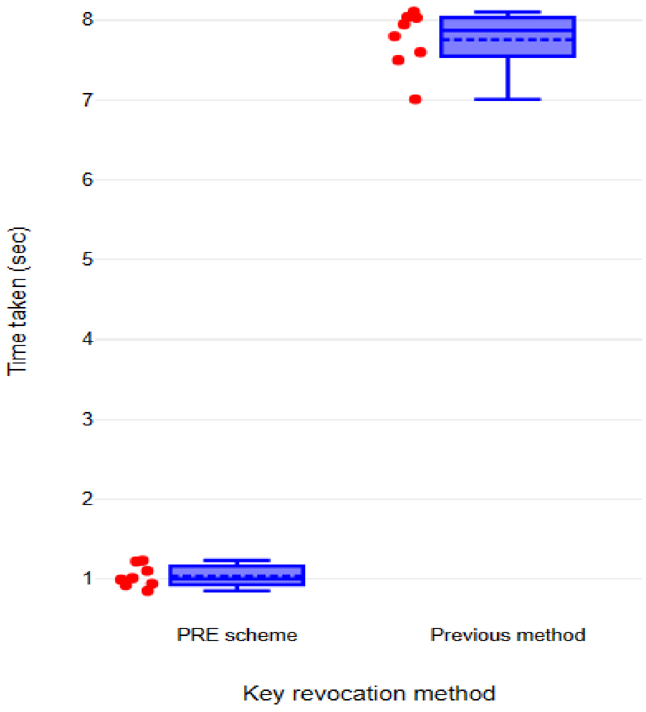


Figure 4-22: Whiskers-box diagram of mean time taken for re-encryption comparison: PRE vs previous method

The results of the evaluation phase show that the evaluation of the execution time and the data retrieval delay provided an answer to RQ1, as one can evidently conclude that the proposed model improved the efficiency of the ABE operations in IoT applications and improved the data retrieval time by 32% when compared with the old scheme. CPU load and memory consumption evaluation results show that the new model's ABE scheme constructed by Goyal et. al (2006) is more efficient than the previous scheme in terms of hardware usage. The evaluation also showed that the proposed PRE scheme's effect in terms of efficiency is that it is significantly more efficient than the old key revocation method as it improved the re-encryption time by 87%. One can also conclude that using the proposed model will enhance the security in IoT applications in case of key revocation or compromise.

5 Discussion and Future work

In this chapter, the author presents a discussion about the study results and limitations, the quality and effects of the artefact, and future work.

5.1 Summary

The thesis aims to reduce the number of security and privacy issues in IoT in lightweight applications and extend the functionalities of a previous framework that showed some research gaps. To attain this aim, the author set out some objectives to achieve the goal of the thesis, which was to construct a model for IoT applications based on ABE and PRE.

Thus, there were two objectives set out, namely:

1. To enhance the efficiency of ABE in an IoT framework
2. To enhance the security of the IoT framework by including an efficient key revocation method for ABE

To investigate how ABE and PRE apply in IoT, the author conducted a document review about existing IoT frameworks, ABE schemes, and key revocation schemes. The author reviewed the IoT middleware architecture proposed by Sicari et al. (2020), the WBAN architecture introduced by Sanchez et al. (2014), and the MQTT architecture proposed by Singh et al. (2015). The articles were filtered in such a way that only ABE schemes suitable for lightweight IoT applications like home, healthcare, and retail were included. Furthermore, the selection of the frameworks was done by filtering the frameworks that are considered state of the art.

From the document reviews, the author was able to identify some gaps in previous architectures. For instance, it was observed that all the three IoT frameworks gather data from heterogenous sources like but none of them checked for source validity, accuracy, or quality of data except the framework proposed by Sicari et al. (2020). However, the framework proposed by Sicari introduces data retrieval delay, and does not implement an efficient key revocation mechanism. The architecture proposed by Sanchez et al. (2014) implements a lightweight CP-ABE scheme that does not introduce a significant delay, but on the other hand, the scheme is not flexible, and the access policies does not offer a reasonable degree of expressiveness. While Singh et al.'s implementation does not implement a database to store IoT data, thus, old data cannot be retrieved, moreover, the architecture does not include a method for key revocation (2015). The thesis' proposed model uses the NOS middleware and adds new methods and components that improves the efficiency and security of the ABE operations. The new components and functionalities introduced in the thesis were identified as requirements to the NOS middleware due to the challenges and research gaps taken from Sicari et al.'s (2020) research. Furthermore, the author chose to extend the functionalities of NOS framework due to its modularity, and its unique ability of combining security, privacy, and quality in a framework that is state of the art (Sicari et al., 2020).

In the document review, the author noted that one of the biggest challenges in IoT is implementing a standardized framework that can be used in all applications (Rizzardi et al., 2016). That is because IoT has diverse applications that can require either heavy computations, such as factories, smart cities, and worksites, or require light computations, such as home applications, healthcare, and businesses. To achieve the purpose of the research, the author chose to use the KP-ABE scheme constructed by Goyal et al. (2006) due to its efficiency in terms of ciphertext size, private key size, and computation time in decryption and encryption as theoretically proven by Goyal. While the old NOS middleware used the CP-ABE scheme constructed by Bethencourt et al. (2007) which when compared to the mentioned KP-ABE scheme performs more exponential operations per attribute during encryption and decryption, which results in more computation time as proved by the evaluation results.

To achieve the second objective, the author performed a document review and noted that PRE is the most suitable method to be used after key revocation. This is because it does not perform naïve re-encryption methods, such as sending the encrypted ciphertext back to the encrypting party to decrypt it and encrypt it using another key. Instead, a re-encryption key is generated based on some rules which is the minimal set of attributes, which produces a ciphertext without seeing the underlying plaintext, thus, achieving efficiency, and scalability (Yu et al., 2010). It was observed that the three models reviewed in the thesis do not mention an efficient method for re-encrypting old ciphertexts like PRE. Therefore, to enhance security and efficiency, the thesis's proposed model uses a PRE scheme for key regeneration, and re-encryption.

5.2 Study limitations

The thesis used the DSR methodology proposed by Johannesen and Perjons (2014) to develop the proposed artefact. The result of the thesis as per Johannessen's classifications is a model of an IoT system that applies ABE in the process of data transfer, storage and encryption. In DSR, a model is a prototype or design of a system that is composed of a group of constructs that have certain functions, the model can then be instantiated based on the different factor affecting its implementation in a certain environment (Johannessen and Perjons, 2014). More in-depth research is needed to identify the best conditions for implementing the system, such as the version of the Raspberry Pi, the type of the network, the request handling module devices, and the used programming languages.

The constructs of the model were built and integrated in such a way that the model can be instantiated, and the system can be implemented in an IoT environment of the choice of the system developer, but some components were not implemented as they were not a part of the scope of the thesis, like the request management module, and the IoT data producers' module. To demonstrate the components using real IoT nodes was a challenging task, rather, the concept of the data producers was demonstrated by using datasets from real IoT smart home testbed. Using real IoT nodes and a request management module would differ in real-time data retrieval delay if the processes were dynamic, but the purpose of the research was to compare the data retrieval delay with that of the old model according to the implemented ABE modules by keeping the experiment factors constant in both models. Evaluating data retrieval by including a request management module like a publish/subscribe broker would produce results which are relative to the implemented modules, in other words, the amount of delay will only differ based on the underlying functionalities of the implemented components which are the ABE scheme, the PRE scheme, and the data storage units by keeping the experiment factors constant, which is what the study compared.

The policies and attributes tested in the KP-ABE scheme were not created using a policy enforcement framework. Rather, the author designed an illustrative case that demonstrates ABE in the proposed model. Implementing a decision logic to produce the access policies is something that can add knowledge to the study. Still, finding the most suitable programming languages and compilers to implement in the IoT device is a challenging task as it is very application specific.

5.3 Ethical and Social aspects

5.3.1 Contributions

Since we are in an era where ethical data handling is a crucial issue for data handling, the model uses encryption which maintains data confidentiality, integrity, and enhances data availability. Only the authorized users with the corresponding criteria can access and modify data, hence, the model helps solve the privacy issues mentioned in the thesis which will lead to an increase in IoT users. Moreover, one of the non-functional requirements decided for the model is ethicality. The model adheres to ethical norms by promoting security goals such as confidentiality and integrity. The model may be shared with people in related industries such as academia and IT industries.

5.3.2 Quality and effects

The proposed artefact fulfills all functional and non-functional requirements identified in the document review phase of this study and the gaps identified in the case study. The requirements that the model meet include applying efficient ABE and adding an extra layer of security by applying PRE. The security model is flexible, scalable, and easy to implement. With this, IoT developers may quickly adapt the model to their existing IT and security infrastructures. Furthermore, the model reduced the data retrieval delay by 32% and re-encryption time by 87%. This factor implies that IoT applications will save on computational costs.

5.4 Conclusion

After the construction, demonstration, and evaluation of the artefact, the results show that the aim of reducing the security and privacy issues in IoT application has been achieved. Also, the two objectives of the research have been met. Investigating how to apply ABE and PRE in the IoT model showed how beneficial they are and led to the final proposed artefact. The practice of using PRE has proven to have a crucial impact on reducing the security issues and enhancing the efficiency in IoT applications based on ABE, which was the aim of the research. In other words, combining ABE and PRE complements the integrity and availability of data with efficiency, which also meets the research aims.

The result of the DSR is a model artefact, the model according to DSR is classified as a prescriptive model that prescribes the components of the IoT security architecture that will implement ABE and PRE. The model consists of three entities, which are: NOS middleware, TA, and data consumers. The components of the NOS middleware are raw data database, data analysis modules, KP-ABE module, PRE module, attributes version list database, PRE keys database, and attributes database. The components of the TA are: KP-ABE module, PRE module, policies database and PRE keys database. NOS platform is deployed on a laptop that installs Ubuntu and the behavior of data consumers are emulated by means of another Ubuntu operating system installed on the same device. The model components were demonstrated using data from real-world smart home testbed by a computer simulation, then, the model was evaluated using an ex-ante approach on a computer simulation.

5.5 Future work

More future research applying DSR could be done to investigate how to use the proposed artefact to produce an instantiation. This implies that the author shall find a setting where an IoT security model based on ABE is needed, the setting can be: A retail environment, an office, or a home that uses IoT. The type of the IoT application and its usage shall be identified before instantiating the model, then, the development steps presented in the thesis shall be followed and adapted based on the setting's needs. A setting where the instantiation can be produced is a home or an office where a web application can be developed.

References

Al-Dahhan, Shi, Lee and Kifayat (2019). Survey on Revocation in Ciphertext-Policy Attribute-Based Encryption. Sensors, 19(7), p.1695. doi: https://doi.org/10.3390/s19071695.

Atzori, L., Iera, A., &Morabito, G. (2011). SIoT: giving a social structure to the internet of things. IEEE Communication Letters, 15(11), 1193–1195.

Aumann, H.H.; Chahine, M.T.; Gautier, C.; Goldberg, M.D.; Kalnay, E.; McMillin, L.M. Revercomb, H.; Rosenkranz, P.W.; Smith, W.L.; Staelin, D.H.; Strow, L.L.; and Susskind, J. AIRS/AMSU/HSB on the Aqua mission: Design, science objectives, data products, and processing systems. IEEE Transactions on Geoscience and Remote Sensing, 41, 2 (2003), 253–264.

Bethencourt, J., Sahai, A. and Waters, B. (2007). Ciphertext-Policy Attribute-Based Encryption. [online] IEEE Xplore. doi: https://doi.org/10.1109/SP.2007.11.

Boyi Xu, Li Da Xu, Hongming Cai, Cheng Xie, Jingyuan Hu and Fenglin Bu (2014). Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services. IEEE Transactions on Industrial Informatics, 10(2), pp.1578–1586. doi: https://doi.org/10.1109/tii.2014.2306382.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3, 77–101. doi:10.1191/1478088706qp063oa

Chow, S.S.M. (2016). A Framework of Multi-Authority Attribute-Based Encryption with Outsourcing and Revocation. Proceedings of the 21st ACM Symposium on Access Control Models and Technologies. doi: https://doi.org/10.1145/2914642.2914659.

Edemacu, K., Park, H.K., Jang, B. and Kim, J.W. (2019). Privacy Provision in Collaborative Ehealth With Attribute-Based Encryption: Survey, Challenges and Future Directions. IEEE Access, 7, pp.89614–89636. doi: https://doi.org/10.1109/access.2019.2925390.

Goyal, V., Pandey, O., Sahai, A. and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM conference on Computer and communications security - CCS ’06. doi: https://doi.org/10.1145/1180405.1180418.

Guo, B., Zhang, D., Wang, Z., Yu, Z., & Zhou, X. (2013). Opportunistic IoT: Exploring the harmonious interaction between humans and the internet of things. Journal of Network and Computer Applications, 36(6)

Haran, M.; Karr, A.; Last, M.; Orso, A.; Porter, A.; Sanil, A.; and Fouche, S. Techniques for classifying executions of deployed software to support software engineering tasks. IEEE Transactions on Software Engineering, 33, 5 (2007), 287–304.

Hernandez-Castro, J. C., Tapiador, J., Peris-Lopez, P., & Quisquater, J. (2013). Cryptanalysis of the SASI ultra-light weight RFID authentication protocol, [cited 2013 May 20]; available from http://arxiv.org/abs/0811.4257.

Hevner, A.R.; March, S.T.; and Park, J. Design research in information systems research. MIS Quarterly, 28, 1 (2004), 75–105.

Johannesson, P. and Perjons, E., 2014. *An introduction to design science*.

Kabachinski, J. (2005). An Introduction to RFID. Biomedical Instrumentation & Technology, [online] 39(2), pp.131–134. doi:10.2345/0899-8205(2005)39[131: AITR]2.0.CO;2.

Klassi, J. Environmental enhancement of the oceans by increased solar radiation from space. Oceans, 17 (November 1985), 1290–1295.

Li, S., Xu, L.D. and Zhao, S. (2014). The internet of things: a survey. Information Systems Frontiers, 17(2), pp.243–259. doi: https://doi.org/10.1007/s10796-014-9492-7.

Li, M., Lou, W. and Ren, K. (2010). Data security and privacy in wireless body area networks. IEEE Wireless Communications, 17(1), pp.51–58. doi: https://doi.org/10.1109/mwc.2010.5416350.

Liu, C.-W., Hsien, W.-F., Yang, C.-C. and Hwang, M.-S. (2016). A Survey of Attribute-based Access Control with User Revocation in Cloud Data Storage. International Journal of Network Security, [online] 18(5), pp.900–916. Available at: http://ijns.jalaxy.com.tw/contents/ijns-v18-n5/ijns-2016-v18-n5-p900-916.pdf.

Locke, D. (2010). MQ Telemetry Transport (MQTT) V3.1 Protocol Specification http://www.ibm.com/developerworks/library/ws-mqtt.

Manyika, James, Michael Chui, Peter Bisson, Jonathan Woetzel, Richard Dobbs, Jacques Bughin, and Dan Aharon. “The Internet of Things: Mapping the Value Beyond the Hype.” McKinsey Global Institute, June 2015. p.3.

March, S., and Smith, G. Design, and natural science research on information technology. Decision Support Systems, 15, 4 (1995), 251–266.

Picazo-Sanchez, P., Tapiador, J., Peris-Lopez, P. and Suarez-Tangil, G. (2014). Secure Publish-Subscribe Protocols for Heterogeneous Medical Wireless Body Area Networks. Sensors, 14(12), pp.22619–22642. doi: https://doi.org/10.3390/s141222619.

Rasori, M., Perazzo, P. and Dini, G. (2020). A lightweight and scalable attribute-based encryption system for smart cities. Computer Communications, 149, pp.78–89.

Rasori, M., La Manna, M., Perazzo, P. and Dini, G. (2022). A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things. IEEE Internet of Things Journal, pp.1–1. doi: https://doi.org/10.1109/jiot.2022.3154039.

Rizzardi, A., Miorandi, D., Sicari, S., Cappiello, C. and Coen-Porisini, A. (2015). Networked Smart Objects: Moving Data Processing Closer to the Source. Internet of Things. IoT Infrastructures, pp.28–35. doi: https://doi.org/10.1007/978-3-319-47075-7_4.

Rose, K., Eldridge, S. and Chapin, L., (2015). The internet of things: An overview. The internet society (ISOC), 80, pp.1-50.

Sicari, S., Rizzardi, A., Miorandi, D., Cappiello, C. and Coen-Porisini, A. (2016). A secure and quality-aware prototypical architecture for the Internet of Things. Information Systems, 58, pp.43–55. doi:10.1016/j.is.2016.02.003.

Sicari, S., Rizzardi, A., Miorandi, D., Cappiello, C. and Coen-Porisini, A. (2016). Security policy enforcement for networked smart objects. Computer Networks, 108, pp.133–147. doi: https://doi.org/10.1016/j.comnet.2016.08.014.

Sicari, S., Rizzardi, A., Dini, G., Perazzo, P., La Manna, M. and Coen-Porisini, A. (2020). Attribute-based encryption and sticky policies for data access control in a smart home scenario: a comparison on networked smart object middleware. International Journal of Information Security. doi: https://doi.org/10.1007/s10207-020-00526-3

Selar, G.D. and Apoorva, P. (2017). Comparative study on KP-ABE and CP-ABE algorithm for secure data retrieval in military network. [online] IEEE Xplore. doi: https://doi.org/10.1109/I2C2.2017.8321816.

Simon, H. (1969) The Sciences of the Artificial. Cambridge, MA: MIT Press.

Singh, M., Rajan, M.A., Shivraj, V.L. and Balamuralidhar, P. (2015). Secure MQTT for Internet of Things (IoT). [online] IEEE Xplore. doi: https://doi.org/10.1109/CSNT.2015.16

Snyder, H. (2019). Literature Review as a Research methodology: an Overview and Guidelines. Journal of Business Research, [online] 104(104), pp.333–339. doi: 10.1016/j.jbusres.2019.07.039

Subhas Chandra Mukhopadhyay (2014). Internet of Things Challenges and Opportunities. Cham Springer International Publishing

Wang, X., Zhang, J., Schooler, E.M. and Ion, M. (2014). Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT. 2014 IEEE International Conference on Communications (ICC). [online] doi: https://doi.org/10.1109/icc.2014.6883405

Weigand, H., Johannesson, P. and Andersson, B. (2021). An artifact ontology for design science research. Data & Knowledge Engineering, 133, p.101878. doi: https://doi.org/10.1016/j.datak.2021.101878

Yao, X., Chen, Z. and Tian, Y. (2015). A lightweight attribute-based encryption scheme for the Internet of Things. Future Generation Computer Systems, 49, pp.104–112. doi: https://doi.org/10.1016/j.future.2014.10.010

Yu, S., Wang, C., Ren, K. and Lou, W. (2010). Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. [online] IEEE Xplore. doi: https://doi.org/10.1109/INFCOM.2010.5462174