IoT

Article

Secure Adaptive Context-Aware ABE for Smart Environments

Saad Inshi [1], Rasel Chowdhury [1,*], Hakima Ould-Slimane [2] and Chamseddine Talhi [1]

[1] Department of Software Engineering and Information Technology, École de Technologie Supérieure, Montréal, QC H3C 1K3, Canada
[2] Département de Mathématiques et d'Informatique, Université du Québec à Trois-Rivières, Trois-Rivières, QC G9A 5H7, Canada
* Correspondence: rasel.chowdhury.1@ens.etsmtl.ca

Abstract: Predicting context-aware activities using machine-learning techniques is evolving to become more readily available as a major driver of the growth of IoT applications to match the needs of the future smart autonomous environments. However, with today's increasing security risks in the emerging cloud technologies, which share massive data capabilities and impose regulation requirements on privacy, as well as the emergence of new multiuser, multiprofile, and multidevice technologies, there is a growing need for new approaches to address the new challenges of autonomous context awareness and its fine-grained security-enforcement models. The solutions proposed in this work aim to extend our previous LCA-ABE work to provide an intelligent, dynamic creation of context-aware policies, which has been achieved through deploying smart-learning techniques. It also provides data consent, automated access control, and secure end-to-end communications by leveraging attribute-based encryption (ABE). Moreover, our policy-driven orchestration model is able to achieve an efficient, real-time enforcement of authentication and authorization (AA) as well as federation services between users, service providers, and connected devices by aggregating, modelling, and reasoning context information and then updating consent accordingly in autonomous ways. Furthermore, our framework ensures that the accuracy of our algorithms is above 90% and their precision is around 85%, which is considerably high compared to the other reviewed approaches. Finally, the solution fulfills the newly imposed privacy regulations and leverages the full power of IoT smart environments.

Keywords: context aware; attribute-based encryption; privacy; security; machine learning; smart environment

Citation: Inshi, S.; Chowdhury, R.; Ould-Slimane, H.;Talhi, C. Secure Adaptive Context-Aware ABE for Smart Environments. *IoT* **2023**, *4*, 112–130. https://doi.org/10.3390/iot4020007

Academic Editor: Amiya Nayak

Received: 12 December 2022
Revised: 11 March 2023
Accepted: 16 April 2023
Published: 20 April 2023

1. Introduction

Security and privacy have become a headache for the industry, which has to cope with the current technological trends as well as the future vision of 6G networks that will impact every interconnected device by including intelligent connections. Context awareness and encryption can be used to tackle these challenges coming from the industry. Context awareness is a term used to represent the case where computers and embedded devices sense and react according to changes in their environment. First introduced by Schilit and Theimer [1] in 1994, a context-aware system acquires, understands, and recognizes the context, and then takes an action according to that precise context. Context awareness evolved from desktop applications, web applications, mobile computing, and pervasive/ubiquitous cloud computing to the Internet of Things (IoT) over the last few years. Moreover, 6G networks are continuously evolving to match the needs of the future IoT smart, self-adoptive applications. Currently, one billion IoT devices are connected machine-to-machine (M2M), and these interrelated devices are provided with unique identifiers and the ability to transfer data over a network with the improved efficiency, accuracy, and economic benefit of IoT environments. Therefore, new applications and business models of the future IoT require new performance criteria, such as big data, multidevice functionality, limited access control, security, privacy, and regulations.

Besides the massive scale of connected end devices, new multidevice technologies have advanced, making identity and access management more complex, while bringing new security and privacy challenges. Smartwatches are good examples of devices that complement smartphones, with the capabilities for checking time, messages, emails, notifications, and many more functionalities with easier accessibility. Another example is vehicles equipped with sensors and internet access that connect to other devices, both inside and outside the vehicle, to provide additional benefits such as traffic alerts and emergency assistance. Section 5 covers more of these solution examples and use-cases in detail.

While the success of all these technological advancements is obvious, the complexity that they add to both identity and access management is crucial. The massive volume of data generated by these devices raises the need for a context-aware, adaptive access control solution to avoid any misuse or leak of confidential information to unauthorized parties.

Moreover, this rapid growth will evolve the exchange of big data to build these smart and self-conscious autonomous environments. According to a study by Zaslavsky et al. [2], it is expected that the total amount of data on earth will reach up to 35 ZB in 2022. Thus, big data becomes more challenging and raises new IoT and 5G requirements for higher levels of access control, context awareness, privacy, and security.

In addition to the big data challenges, the protection of personal data becomes very relevant for the adoption of these technologies. Therefore, it is critically important to properly understand the main aspects of current regulations, which have an impact on 5G and IoT security. The new regulations are mainly proposed to enhance, unify, and protect the data of each individual. The European Union (EU) has formulated and planned to implement and enforce a new general data protection regulation (GDPR) [3]. The GDPR is anticipated to protect the export of personal data within and outside the EU. Furthermore, the United States is working to protect customers, maintain competitions, and advance organizational performance by forming the Federal Trade Commission (FTC) and the Federal Information Security Management Act (FISMA). The FTC and FISMA follow dynamic and effective law enforcement, with the principle mission of protecting consumer privacy [4,5]. All of these regulations impose that service providers cannot collect personal information that is not required, while data collection, storage, and processing have to be conducted securely. A service provider (SP) should keep the data of the user only during the business period, and the user is the one who has the right to grant access to their data. Hereby, the continuous success and future development of 6G and the IoT will depend on the ability to adopt and comply with these regulations.

Accordingly, as will be discussed in the literature section, several studies have covered many context-aware aspects [6–16], of which two main aspects are still challenging: The first aspect is the lack of a context-awareness broker that can manage the complex identities of interconnected sensors from different devices. In such situations, context-aware policies will play the main role in deciding what and when data need to be processed or shared, and much more. In addition, the current smart IoT environments need such context-awareness brokers as central points of control, as different middleware solutions developed by different parties will be employed to connect to sensors for collecting, modelling, and reasoning context information.

The second aspect is updating consent based on the context information. While the data aggregated by the sensors need to be shared with the authorized providers or observers, consent is not fixed and can be updated so that data are not shared in certain contexts. In our study, as shown in Figure 1, our main contexts are activity, identity, time, and location. Furthermore, we collected data from multiple devices, such as watches, cars, and tablets, which are viewed as inputs for the contexts. For example, consent can be updated so that data are shared when switching between devices or when switching between users, or so that data based on location and time (in meetings, at work, and at home) are not shared. Furthermore, consent can be updated to share data again based on events, such as an insurance company needing to restore data to determine what caused an accident. Therefore, to effectively perform this aspect, we need an intelligent, dynamic adaptation of contexts, which can be achieved through smart-learning techniques. In such situations, machine-learning techniques will take the context information as inputs to learn models that are able to adapt the consent accordingly.

Therefore, the context broker will coexist with different middlewares by managing the complex identities of interconnected multidevice sensors for different providers. In these situations, context-aware policies will play the main role in deciding what and when data need to be processed or shared, and much more. In addition, the current smart IoT environments need such context brokers to be central points of control, as different middleware solutions developed by different parties will be employed to connect to sensors and collect, model, and reason context. When a subscriber device wants to send data to the business application or application server, it has to authenticate itself to the broker to obtain the decryption key from the KMS. When the service provider (SP) wishes to decrypt the data, it has to authenticate itself to the broker to obtain the decryption key from the KMS. The broker will check the context and the access privileges for that data and the SP. If the SP is authorized, it will be able to see the data. The details of these processes are shown in Section 3.

In addition to all of the above challenges, security is a key requirement for the evolution of the new automated smart IoT environments that provide security services and identity management. This growth raises the need to protect data exchanged between different entities by many methods (e.g., authorization, automated access control, and data encryption). In this scope, many encryption-based access control schemes have been proposed and adopted in IoT environments, starting with symmetric key encryption and moving to the public key infrastructure (PKI), which provides a data signature to ensure integrity and session keys to ensure confidentiality. Still, PKI requires a huge infrastructure (certifying authorities, registration authorities, repositories, archives, and end entities) to manage and maintain the certificates [17]. Goyal et al. [18] introduced an asymmetric encryption technique called attribute-based encryption (ABE), which is one of the most recent access-control-based encryption schemes. This technique allows for the definition of fine-grained access control policies for data privacy and security. Yet none of the existing works fulfill our requirements for context-aware, smart, adaptive environment use-cases.

Ultimately, smart context-aware encryption is the key solution to new secure smart systems evolution. The present novel solution discloses a context-aware, adaptable, intelligent, and lightweight security solution. The solution extends our previous work on LCA-ABE [12] to deliver our adaptable data consent, automated access control, and secure end-to-end communications between different users, network operators, and service providers by aggregating, modelling, and reasoning context information and then updating consent accordingly in an autonomous way. Finally, the solution fulfills the newly imposed privacy regulations, considering the 5G technologies, leveraging the full power of IoT security, multi environments, and access control of big data.

The main contributions of this research are as follows:

- Proposes a context-aware, adaptable, intelligent, and lightweight security solution.
- Achieves the novel intelligent dynamic creation of context-awareness policies through smart-learning techniques.
- Provides a context-aware dynamic encryption model by leveraging attribute-based encryption (ABE).
- Formulates the ABE context-awareness policies based on machine-learning techniques.
- Presents a solution that fulfills the newly imposed privacy regulations.

We organized the article into different sections. Section 2 gives a brief idea of the background and related research in the field of context-aware systems, machine learning, and ABE. We explain the architectural design in Section 3, and in Section 4 we explain the formal definition of the problem. In Section 5, we have the use-cases, implementation, and evaluation. The conclusion and future work are in Section 6.

2. Background and Literature Review

In the IoT domain, there is lots of research being performed in the field of context-aware security and attribute-based encryption. In this section, we will mainly list out the research specific to adaptive context awareness and the feasibility of using ABE in the IoT domain.

2.1. Adaptive Context Awareness

The context-awareness policies vary depending on each domain's requirements, such as mobile computing, web applications, and, recently, IoT smart environments. One of the leading mechanisms for defining context types in IoT environments is presented in [19], where they have two main categories: The primary context and the secondary context. The primary context represents any information retrieved directly from sensors, such as location data from a GPS sensor, user identity based on a SIM card, time read from a device clock, or activity from a smartwatch sensor. The secondary context represents any information that can be computed using the primary context, such as predicting the user's activity based on the user's calendar or predicting the user's location based on an image retrieved from a map service provider. Figure 1 presents a context-awareness taxonomy for IoT environments. According to the taxonomy, we can split the context awareness into four main steps, named the context life cycle [20]. Each phase of the life cycle has its own challenges. The challenges of the acquisition step are related to the context-aware system's capability to support the registration of new data sources (i.e., data providers, sensors, and devices) and its technical issues, as well as the network communication used to acquire the data source information. The modelling and reasoning phases hold the same challenges because these phases strictly depend on each other.

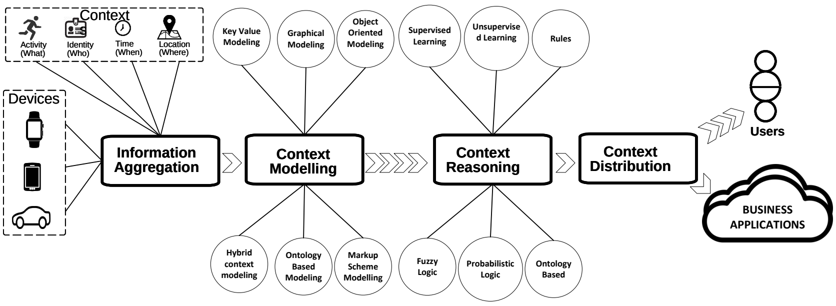

Figure 1. Context awareness Life Cycle.

In today's era of big data and smart environments, many researchers raise the essential need for context awareness and dynamic access policy management to protect resources from unauthorized access. In [9,13–16], the authors proposed different automatic adaptive access policy specification frameworks for IoT environments based on a rule-based method. This research conducted an interesting dynamic accessibility method, but did not provide an adequate machine-learning method to support such adaptive access control.

Context-awareness aspects along with machine-learning techniques have been successfully used in different domains, including mobile cloud computing, E-health, smart homes, and IoT smart autonomous environments where most of these systems rely on intelligent context-aware applications. The researchers in [6,21] investigated various popular machine-learning classification techniques, such as decision trees (DT), k-nearest neighbour (KNN), random forest (RF), naive Bayes classifier (NB), support vector machines (SVM), and deep learning, for creating smart models for different mobile use-cases. In the mobile cloud computing domain, some studies, such as [7], have analyzed the context of the device's operations and their related security aspects. Furthermore, they have developed an agent-based adaptive system that uses machine-learning algorithms to enable the optimization of services on the mobile device and secure the allocation of tasks in conventional cloud resources. Moreover, the idea of using ML and context awareness is investigated in many studies, such as [8,10], to provide the appropriate services to users in E-health and smart homes.

In summary, the above-discussed ML and context-awareness techniques allow for the definition of fine-grained, context-aware access control policies for data privacy in different domains. Yet none of the existing works fulfill our requirements for the automated creation of context-aware policies and the automated encryption according to these auto-generated policies. Therefore, in this work, we have experimentally tested more use-cases to automatically analyze the user behaviour according to different context information and the utilization of their smart-device datasets. All use-cases will be discussed in the experimentation and evaluation section.

2.2. Attribute-Based Encryption

The authors in [18,22] proposed an encryption technique that allows access control based on attributes. According to various studies, ABE is one of the best techniques to have access control along with data security. ABE is a public key encryption technique, which utilizes attributes and policy to encrypt and decrypt the data as well as access control. Attributes can be basically anything, such as postal code, departments, locations, services, etc. There are two types of ABE, cipher-text policy ABE (CP-ABE) [22] and key policy ABE (KP-ABE) [18].

There has been a lot of research conducted in various domains with ABE. The authors in [23–25] performed different types of analyses regarding the feasibility, performance, and survey of ABE in the cloud, smart devices, and smart environments. Ambrosin et al. [26] provided a comprehensive study in terms of resource utilization and execution time for CP-ABE and KP-ABE in constrained devices. From their experimentation, it is clear that ABE can be incorporated in resource-constrained devices. In [27,28], the authors proposed that ABE can be used to secure patient data before storing them on the cloud where only the specific personnel can view the patient records. The authors in [29] also showed that it is important to encrypt the patient data collected from different sources in the hospital before sending them to the cloud. The authors in [30] implemented a framework using OpenHAB to secure the data generated in the smart home before being shared with the cloud service providers. In [31], the authors studied the feasibility of the ABE in a body sensor network and proved that it has very good prospects in the domain of smart devices. The authors in [32] proposed online/offline attribute-based proxy re-encryption for smart phones to encrypt the data before they are transferred to the cloud. They also showed the performance and feasibility of using their ABE scheme in smart devices such as cellphones. In [33], the authors proposed a CP-ABE offloading technique where the authors performed partial encryption on the resource-constrained devices, and the actual encryption in the cloud where there are enough resources.

2.3. Summary of the Literature Reviews

As shown in Table 1, previous studies have covered many mobile and IoT context-aware aspects for different domains. Many technical limitations need to be addressed in order to meet with the new IoT context-aware, adaptable, intelligent, and lightweight security solution requirements. Therefore, we have classified the main reviewed work according to our research objectives. Furthermore, many other research and industrial aspects in the future of IoT applications are still challenging to fulfill, such as the newly imposed privacy regulations and the emergence of new multiuser, multiprofile, and multi-device technologies. Above all the mentioned aspects, security is a key requirement for the evolution of new automated smart IoT environments. Therefore, context-aware encryption is our proposed solution for overcoming some of the reviewed security limitations and opening up promising future directions for both academia and industry professionals.

Table 1. Comparison of existing related work.

Reference	Domain	Rule Base	Machine Learning Algorithm	Context aware	Dynamic	Multi Users	Privacy	ABE
BehavDT [6]	Mobile	✗	Decision Tree	✓	✓	✗	✗	✗
ACAO [7]	Mobile Cloud Computing	✗	Naïve Bayes (NB), Decision Tree, Random Forest	✓	✓	✗	✓	✗
SMAF [8]	E-health	✗	Grey Model (GM)	✓	✓	✗	✗	✗
[9]	IoT	✓	✗	✓	✓	✗	✓	✗
AppsPred [10]	Smart home	✓	Random Forest	✓	✓	✗	✗	✗
HybridGuard [11]	Hybrid Mobile	✓	✗	✓	✗	✗	✗	✗
LCA-ABE [12]	Mobile	✓	✗	✓	✗	✗	✓	✓
Our work	Smart Environment	✓	Artificial Neural Network Markov Chain	✓	✓	✓	✓	✓

3. Architecture

In this research, we propose a new approach for fine-grained privacy and confidentiality using context awareness and ABE for access control and data security. We are extending our previous work [12] and adding extra components to make the framework dynamic and adaptive with the context. Figure 2 shows our architecture for smart-environment systems.

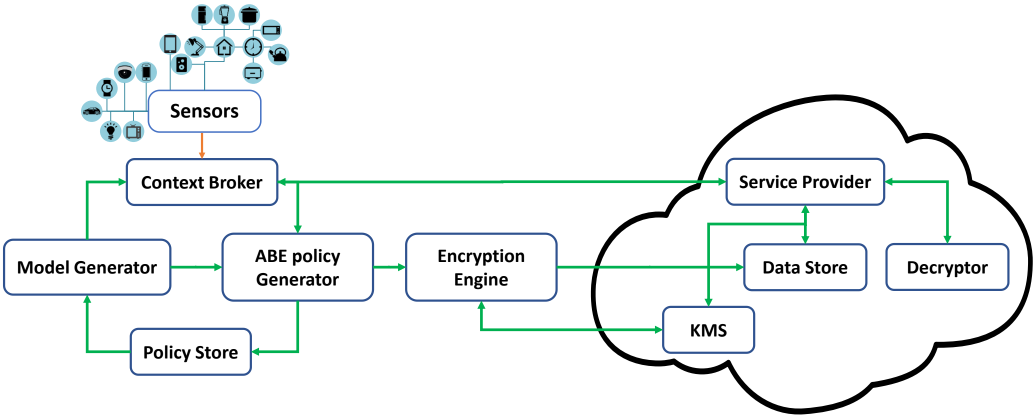

Figure 2. Proposed Architecture.

The architecture is divided into multiple modules, which are as follows:

- The **context broker** is the entry point for all the data coming from the sensors and different services, and it also acts as a medium for the services to access the sensors and service data. The *context broker* has two functionalities:
  - When the *context broker* receives data, it generates a context based on the location, time, activity, etc., as attributes, and transfers the data and the attributes to the *ABE policy generator* for further processing.
  - When the *context broker* receives a request from a *service provider*, it retrieves the context from the environment and uses the classification algorithm to decide whether the *service provider* has the rights to access the data. If the *service provider* has access, then the *context broker* gives the *service provider* access to the database to access the data; otherwise, it does not. Once the decision is completed, the context and the decision are both sent to the policy storage for future use.
- The **ABE policy generator** uses the attributes received from the *context broker* and uses the model generated by the *model generator* to generate a CP-ABE policy, and then forwards the data and the policy to the *encryption engine*. Furthermore, it sends the policy to the *policy store* as a dataset entry for future use.
- The **encryption engine** encrypts the data using the policy received from the *ABE policy generator* using CP-ABE and saves it in the database.

- The **policy store** is a database that stores the context and decision for the service providers, as well as the ABE policy generated for each context, all of which will be used by the *model generator.*
- The **model generator** periodically collects the data stored in the *policy store* to generate machine-learning models, which will be used by the *context broker* and *ABE policy generator*. For the *context broker*, it uses the naive Bayes classification model, while for the *ABE policy generator*, it generates a Markov chain.
- The **key management system (KMS)** generates and stores all the necessary keys, such as the master key, public key, and private key, for the whole ecosystem.
- The **decryptor** is a decryption module used by the service providers to decrypt the data they requested. The *decryptor* will try to decrypt the data using the secret key from the service provider, and if the service provider has access to the data, then it will return the data, else it will return denied access to the data.
- The **data store** is storage in the cloud, or can be in the device itself, where the data are stored after the encryption process in complete.
- The **service provider** is the application or cloud service that the user has subscribed to.

Figures 3 and 4 explain the sequence of operations for our framework. Figure 3 shows the sequence diagram for the data life cycle when it arrives to the context broker from the sensor. Figure 4 shows the steps of operation being performed when a service provider wants to access a sensor's data.

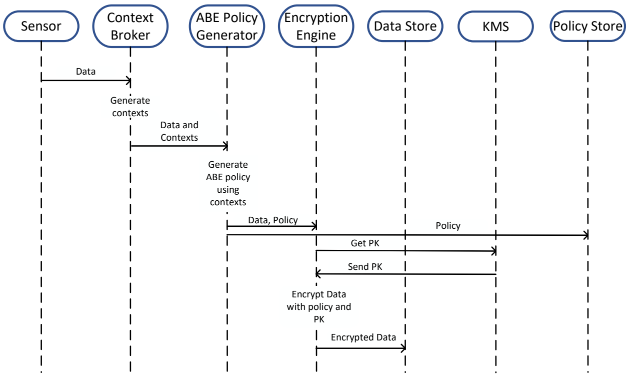

Figure 3. Sequence Diagram for sensors generating data.

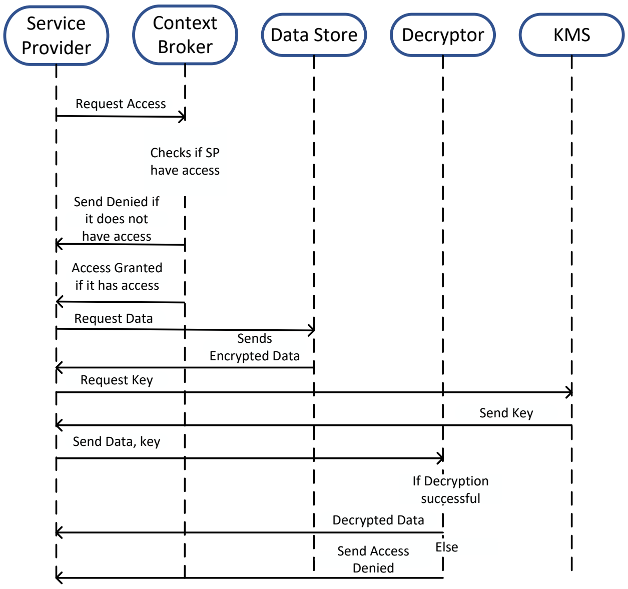

Figure 4. Sequence Diagram for Service Provider requesting data.

4. Problem Definition

IoT smart applications must be adaptable, intelligent, and secure in order to fulfill the newly imposed privacy regulations, defining and updating consent according to the user's context information in autonomous ways. Furthermore, the execution of the context-aware policies for IoT services and devices must be dynamic, lightweight, and platform-specific. In order to tackle the dynamic nature of the IoT environment, we need to solve the autonomic access control problem and formulate an adaptive policy that can be generated on the basis of context. We present a formal definition of our research problem for context awareness and ABE. In this section, we perform the derivation of formulas for context-aware classification and ABE policy generation. In this research, the problem definition for the context-aware ABE is conducted in two phases, namely:

- The applications need to access the data, but due to security reasons, that needs to be controlled based on the permission, context, source, and application details. The access granted to the application is either "Allow" or "Deny", which is a binary classification problem wherein the permission, context, source, etc., can be viewed as features, and based on these features, we need to classify whether the application will be granted access or not.
- The context provided by the system needs to be transformed into ABE policy automatically, using operators such as "AND" or "OR" and contexts such as "time", "location", etc., which can be viewed as prediction problems. We need to predict the best possible operators and contexts for encrypting the data that can satisfy the user's behaviour.

4.1. Context-Aware Access Control

In a smart environment, there are applications and cloud services that require ambient data generated by the sensors and other applications in order to provide the user with a better experience and ease of use. However, these data need to be shared with the applications and cloud services based on the user permissions.

In this research, we extended the context-aware policy definition of LCA-ABE [12] for the formulation of our context-aware access control. An application or cloud service *App* contains a list of information, such as name, class, visibility, and API, which we denote as *App* = (name, class, visibility, API). Permission of an *App* *P*, which is *P* = (name, resources, securityLevel), where name is the name of the permission, resources can be personal data, calendar, camera, etc., and securityLevel is the level of security of the permission, such as normal or dangerous. The context c is the circumstances during which the data are generated from the sensors and application. The context can be the time when the data are generated, location of the data, activity of person using the device, etc. The context is represented by c = (time, location, activity).

Based on our context-aware policy definition, a context-aware rule for a fitness application can be written as $\text{Allow}_{gps}(\text{Fitness}) \leftarrow (\text{gps} \in \text{Fitness.APIs}) \wedge (\text{P.resources} = \text{gps}) \wedge (\text{c.activity} = \text{exercise}) \wedge (\text{c.time} = \text{7am}) \wedge (\text{c.location} = \text{park})$. Another example, $\text{Deny}_{notifications} \leftarrow ((\text{notification} \in \forall\ \text{APIs}) \wedge (\text{notification} \neq (\text{gps} \vee \text{maps}) \wedge (\text{c.activity} = \text{driving})$ will be applied to the user's car, so that all the notifications from the cellphone are blocked except the navigation system. In order to automate the decision process of context-aware policies, the policies were transformed into tables with multiple features that can be used by a machine-learning algorithm, as shown in Table 2. Using a simple classification algorithm, we are able to solve the context-aware control problem. For instance, as shown in [34], if the user has to include all the rules for a device's sensor data for access control, then it will become a hassle and be infeasible, so we need to use an automated system that will grant the access for the device data based on the user's behaviour. So in order to automate this process, we need to use machine learning, where the ML algorithm can automatically grant the access permission and generate the access policy on behalf of the user.

Table 2. Sample Transformation of context-aware policy.

Application			Permission				Context			Source	User	Decision
Name	**Class**	**Visibility**	**API**	**Name**	**Resource Type**	**Security Level**	**Time**	**Location**	**Activity**			
Fitness	Well-being	Background	Get data	GPS	GPS	Medium	700	Park	Jogging	GPS	Bob	Allow
Fitness	Well-being	Background	Get data	GPS	GPS	High	900	Office	Meeting	GPS	Bob	Deny
Map	Navigation	Background	All	All	Notification	Medium	800	Hollywood Boulevard	Driving	Apps	Bob	Deny
Map	Navigation	Background	GPS	GPS	Notification	Medium	800	Hollywood Boulevard	Driving	Apps	Bob	Allow

4.2. Dynamic ABE Policy

The data which are being generated from the different sources need to be encrypted before being stored in the device itself or sent to the cloud. Furthermore, based on the context of the user, the data privacy will change. ABE, which provides encryption of data along with access control to the data, is the most suitable solution for the context-aware environment [35,36]. In order to encrypt data, ABE needs a policy that contains attributes and operators that combine these attributes, e.g., "Bob AND 7AM AND Exercise AND Well-being" is a policy for encrypting the GPS data that makes sure that the fitness application can access the data for that context. However, the catch for this process is to generate the ABE policy dynamically, which is challenging. In order to hurdle the challenge, we introduce a new procedure for the automatic generation of the ABE policies.

Figure 5 shows an example of ABE policy based on the context provided by the system. The system has all the attributes, i.e., the contexts, the only thing missing being the operators, which can be easily determined based on the user's predetermined behaviour. This problem can be viewed as a travelling salesperson problem, where we need to find the best route from beginning to end.

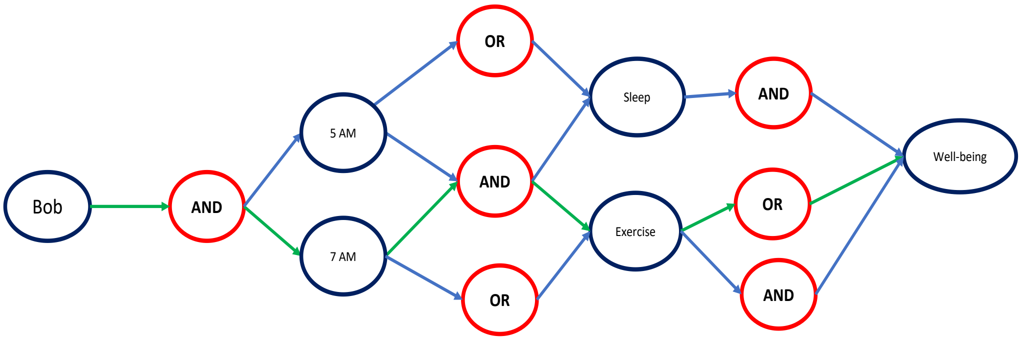

Figure 5. A visual representation of ABE policy.

The application, permission, context, and users, which are attributes denoted by A, are assumed as cities, and the operators 'AND' and 'OR' are the cost for travelling from one city to another. The cost is calculated based on the probability of the operators based on the behaviour of the user, based on the example in Figure 5. For reference, the transition matrix will look like Table 3. The probability of each state will be calculated using $S_n = S_0 \times Q^n$, where S_n is the initial state vector, which is Bob, and Q is the transition matrix to move from state i to state j, as shown in Table 3. In our research, we did not intend to find the path with the best cost, but searched for the longest one because that is where we will find the best attributes for these contexts, which will eventually become the policy for the data's ABE encryption. Using Dantzig–Fulkerson–Johnson formulation, we can find the path without having subtours or loops. So the equation without loops is

$$A_{ij} = \begin{cases} 1 & The\ path\ goes\ from\ A_i\ to\ A_j \\ 0 & Otherwise \end{cases}.$$

Table 3. Transition Matrix.

	Bob		5am		7am		Sleep		Exercise		Well-Being	
	AND	**OR**	**AND**	**OR**	**AND**	**OR**	**AND**	**OR**	**AND**	**OR**	**AND**	**OR**
Bob	0	0	0.3	0	0.5	0	0.02	0.02	0.03	0.04	0.03	0.06
5am	0	0	0	0	0	0	0.4	0.3	0.1	0.05	0.06	0.09
7am	0	0	0	0	0	0	0.15	0.15	0.3	0.2	0.1	0.1
Sleep	0	0	0	0	0	0	0	0	0	0	1	0
Exercise	0	0	0	0	0	0	0	0	0	0	0.4	0.6

By taking $C_{ij} > 0$ to be the cost from attribute i to attribute j, the formulation can be derived using an integer linear programming problem:

$$\begin{aligned} & max \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} C_{ij} A_{ij} \quad where, \\ & \sum_{i=1, i \neq j}^{n} A_{ij} = 1 \\ & \sum_{j=1, j \neq i}^{n} A_{ij} = 1 \\ & \sum_{i \in Q} \sum_{j \neq i, j \in Q} A_{ij} \leq Q - 1 \qquad \forall Q \not\subseteq 1, \ldots, n, Q \geq 2 \end{aligned} \tag{1}$$

In order to solve this NP hard problem, we have utilized a discrete Markov chain to find the best attributes for the policy chain based on the context provided. The Markov chain is a stochastic statistical model, where the future state depends on the past states. In our case, the total number of contexts available in the environment are the states. So, the following equation (Equation (2)) is the classical Markov equation of higher order [37], and by taking as state space the ordered m-tuples of A, it is used in order to find the optimal policy chain of that particular context for encrypting the data.

$$\mathbb{P}(A_n = a_0 A^{n-1} = a_1, \ldots, A^{n-k} = a_k) = max \sum_{i=1}^{k} \lambda_i q_{a_0} a_i \tag{2}$$

where $\sum_{i=1}^{k} \lambda_i = 1$, λ_i is non-negative, k is the total number of attributes, and Q = [q $_{ij}$] is the transition matrix where the sum of the column is equal to one. Using [38], the generalization of [37] as follows:

$$A^{n+k+1} = max \sum_{i=1}^{k} \lambda_i Q_i A^{(n+k+1-i)} \tag{3}$$

With Equation (3), we will be able to find the best path with the highest probability given any number of attributes and form the policy for the ABE encryption. Using Equation (3) and the transition matrix, the result for different states is calculated in the form of Bob 7am (0.3), Exercise (0.3), Well-being (0.6). Then, using the transition matrix and the value, we will find the operators, and the policy will look like Bob AND Time = "7am" AND Exercise OR Well-being.

4.3. Algorithm

We have utilized two algorithms to solve the context-aware-based encryption problem. Algorithm 1 is designed for dynamic access control and Algorithm 2 is for generating the ABE policy for encrypting the data based on the context, where Algorithm 1 is hosted in the context broker and Algorithm 2 is in the ABE policy generator. We have created the initial transition matrix using a probabilistic method. The descriptions of the algorithms are as follows:

Algorithm 1 performs the decision making for the application when it wants to access specific data from the data store. The algorithm takes parameters as shown in Table 2 and uses a trained deep-learning model to take the decision, whether it has access to the data or not, with respect to the current contexts.

Algorithm 1 Access Control

```
Require: Parameters[Application, Permission, Context, Source, User], Trained Model
  Result = Classify the Parameters using Trained Model
  if Result = Allow then
    give access to the data
  else
    return "Denied" to the application or service
  end if
```

Algorithm 2 is called whenever the system generates data. It automatically generates an ABE policy based on the contexts (attributes) provided by the system. The algorithm takes attributes as shown in Table 2, which contains application details, permission of the application, context, data source, and the user who is currently using the device. Furthermore, the algorithm requires a transition matrix that contains the probabilities of each attribute with another attribute. The transform array contains the probability, which corresponds to the operators. The algorithm then uses Equation (2) to calculate the best path for the attribute and stores the value of the path it took to reach the goal. After that, the algorithm uses the values and the transform array to find the operators. Finally, using the path and the operators, it generates the ABE policy for the current context and sends it to the encryption module.

Algorithm 2 Policy Generator

```
Require: Attributes[Application, Permission, Context, Source, User], Transition Matrix M,
  Transform Array T
  Path = Evaluate Equation (2) using Attributes and M
  Operators = Get operators using Path and T
  for All value in Path do
    Find the value of Path[i] corresponding to T
    Append value Operators
  end for
  for All value in Attributes do
    Append Attributes[i] AND Operators[i] to policy
  end for
  return policy
```

5. Experimentation and Evaluation

In this section, we evaluate our framework in terms of resource utilization as well as evaluate the algorithms. We also present four use-cases used during our experimentation. The use-cases represent the different rules that can be used based on the user's predefined context and their representation of the ABE policy.

5.1. Use-Cases

Figure 6 presents our solution's examples and use-cases. The descriptions of the use-cases are as follows:

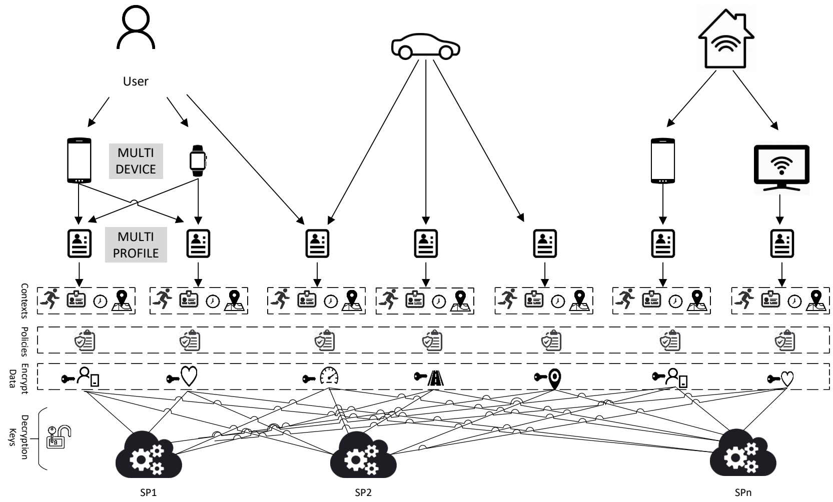

Figure 6. Solution Examples Model.

5.1.1. Scenario 1: Exercising

Bob is going for his regular morning run and is using a fitness application while he exercises. According to normal fitness apps, at the time of installation, Bob was required to allow/deny this app access to some of his smartphone/smartwatch resources. Therefore, using our model, Bob will be able to manage his access policies for while he exercises according to his needs within a specific location and during specific time frames. Examples of the resources that Bob is sharing include location, heartbeat, and accelerator.

Our model defines the application, permissions (resources), and contexts as discussed in Section 4. We assume the following are the permissions that Bob has set to control access to his device:

- Rule 1: The fitness app may have access to heartbeat and accelerator meter sensors while exercising.
- Rule 2: The fitness app may only access the heartbeat and accelerator meter sensors while Bob is at the park.
- Rule 3: The fitness app may only access the heartbeat and accelerator sensors at 6:00 a.m. or 6:00 p.m.
- Rule 4: The fitness app may never share/access the location.

The rules are be represented as follows:

❑ The context-aware access control policy:
  - $\text{Allow}_{FitnessApplication} \leftarrow \text{Activity}_{Sport} \wedge \text{Location}_{park} \wedge \text{Time}_{6am} \vee \text{Time}_{6pm} \leftarrow \text{Data}_{heartbeat,accelerator}$
  - $\text{Deny}_{FitnessApplication} \leftarrow \text{Activity}_{Sport} \wedge \text{Location}_{park} \wedge \text{Time}_{6am} \vee \text{Time}_{6pm} \leftarrow \text{Data}_{location}$

❑ The ABE policy at that specific moment, which are the data collected at 6am for different sensors, is as follows:
  - $\text{Data}_{heartbeat} \rightarrow$ Application = 'Fitness Application' AND Location = 'park' AND Time = '6am' AND Activity = 'Sport'
  - $\text{Data}_{accelerator} \rightarrow$ Application = 'Fitness Application' AND Location = 'park' AND Time = '6am' AND Activity = 'Sport'
  - $\text{Data}_{GPS} \rightarrow$ Application = '$\neq$ *Fitness*' AND Location = 'park' AND Time = '6am' AND Activity = 'Sport'

5.1.2. Scenario 2: Meeting with a Supervisor

Bob is a graduate student who has scheduled regular meetings with his supervisor, Alice. According to Alice and the school regulations, all graduate students must respect some policies regarding access to their device resources while attending meetings, during specific time frames, and in specific locations (offices, labs, or meeting rooms). Examples of the resources that Alice could allow/deny a student access to include the microphone, camera, GPS, and notifications.

We assume the following are the policy rules that Bob sets to fulfill Alice's meeting requirements:

- Rule 1: No app can have access to the voice recorder and camera while in meetings.
- Rule 2: No app can have access to the voice recorder and camera while in a meeting in the meeting room.
- Rule 3: No app can have access to the voice recorder and camera while in a meeting in the meeting room during specific time frames.
- Rule 4: During meetings, no app may ever share/access the location.

Using our model, the rules for this scenario are as follows:

❑ The context-aware access control policy:

  – $\text{Deny}_{Application} \leftarrow \text{Activity}_{Meeting} \wedge \text{Location}_{MeetingRoom} \vee \text{Time}_{\geq 10am} \wedge \text{Time}_{\leq 11am} \leftarrow \text{Data}_{Microphone,Camera,GPS,Notification}$

❑ The ABE policy at that specific moment, which are the data collected at 11am for different sensors:

  – $\text{Data}_{GPS} \rightarrow$ Application = 'All' AND Activity = 'Meeting' AND Location = 'Meeting Room' OR Time = '11am'
  – $\text{Data}_{Camera} \rightarrow$ Application = 'Personal' AND Activity = 'Meeting' AND Location = 'Meeting Room' OR Time = '11am'
  – $\text{Data}_{Microphone} \rightarrow$ Application = 'Personal' AND Activity = 'Meeting' AND Location = 'Meeting Room' AND Time = '11am'

5.1.3. Scenario 3: Driving a Car

Bob has a car insured by a well-known insurance company. Bob does not want to receive any annoying notifications while driving. Furthermore, he does not want to share all of his car's/smart device's resource information with the insurance company while driving. Therefore, Bob defines the attributes and context policies to allow/deny the insurance app access to some of his smart car/smartphone/smartwatch resources according to his consent. Moreover, our model allows Bob to update his consent anytime to share data again based on an event, such as the insurance company needing to obtain the data to determine what caused an accident.

Examples of the resources that Bob could allow/deny the insurance company app access to include GPS tracking, navigation, speed, and hours of driving. We assume the following are the permissions that Bob sets to control access to his devices while driving his car:

- Rule 1: The insurance app has no access to GPS tracking and navigation while driving.
- Rule 2: No app can send/receive notifications while driving.

The rules for this scenario are as follows:

❑ The context-aware access control policy:

  – $\text{Deny}_{Application} \leftarrow \text{Activity}_{Driving} \leftarrow \text{Data}_{Notification}$
  – $\text{Deny}_{Insurance} \leftarrow \text{Activity}_{Driving} \leftarrow \text{Data}_{GPS,Navigation}$

❑ The ABE policy at that specific moment, which are the data collected at 5pm for the GPS sensor:

  – $\text{Data}_{GPS} \rightarrow$ Application = '$\neq$ Insurance' AND Activity = 'Driving' AND Location = 'Peel' AND Time = '5pm'

5.1.4. Scenario 4: Smart Home

Bob is staying home after finishing his job and is watching a TV show with his family on their smart TV. Most applications installed on smart TVs have access to many resources that can leak some private information, such as location and favourite shows, etc. Bob has to allow/deny these apps access to some of their smart home/smart TV resources. Therefore, using our model, Bob will be able to manage their access policies while staying home according to their needs within a range of specific activities (sleeping, watching TV, and eating) and during specific time frames. Examples of the resources that Bob is sharing: location, GPS, and heartbeat.

We assume the following are the permissions that Bob sets to control access to their smart home resources:

- Rule 1: All health apps may have access to heartbeat sensors while sleeping.
- Rule 2: All apps may only access heartbeat and accelerator sensors while Bob is at the park.
- Rule 3: No app may ever share/access the exact location while Bob is at home.

The rules for this scenario are as follows:

❑ The context-aware access control policy:

- $\text{Allow}_{FitnessApplication} \leftarrow \text{Activity}_{Sleeping} \leftarrow \text{Data}_{heartbeat}$
- $\text{Deny}_{Application} \leftarrow \text{Activity}_{WatchingTV} \leftarrow \text{Data}_{Notification}$
- $\text{Deny}_{Application} \leftarrow \text{Activity}_{Sleeping} \leftarrow \text{Data}_{Notification}$

❑ The ABE policy at that specific moment, which are the data collected at 11pm for the GPS sensor:

- $\text{Data}_{heartbeat}$ → Application = 'Fitness Application' AND Location = 'park' AND Time = '11pm'AND Activity = 'Sleeping' OR Activity = 'TV'
- $\text{Data}_{accelerator}$ → Application = 'Fitness Application' AND Location = 'park' AND Time = '11pm' AND Activity = 'Sleeping' OR Activity = 'TV'
- Data_{GPS} → Application = '$\neq$ *Fitness*' AND Location = 'park' AND Time = '11pm' AND AND Activity = 'Sleeping' OR Activity = 'TV'

5.2. Implementation

In this section, we explain the implementation of the main algorithms and dataset for our adaptive context-aware encryption framework

5.2.1. Dataset

The principal role of the system is to automatically grant the access control of the sensor and to generate policy for ABE. For this reason, we have collected data by monitoring Fitbit and a person's cell phone applications (mainly the google map data) for a period of one month. The google map data are collected from the historical information stored in the map history, and the Fitbit data are gathered from the personal historical data from their website. The types of data which are collected are as follows:

- Permissions requested by applications include the name of the sensor it is accessing, the type of resources, and the security level;
- Google Map data while driving, which includes the latitude and longitude;
- Permissions of the services, mainly the GPS, and services requested by the Fitbit;
- Activity tracking, which includes sleep, exercise, swimming, heart rate, SpO2, steps, time in heart-rate zone, and calories burned during that time.

After the collection of the data, we had to removed excess unnecessary features and information such as heart-rate zone, calories burned, etc. Then, we aggregated the data and categorized them into respective features, which is similar to Table 2. Some of the features, such as the source of the data, permission name, and resource type, are correlated, which was not covered in this article. The data for the first dataset are balanced as we manually labelled the data for the access control that will be used by algorithm 1 for classification. The second dataset contains the ABE policy, which we manually created to test whether Algorithm 2 is able to generate the desired policy. This dataset is mainly used to create a transition matrix by using the probabilistic methods, which is similar to Table 3.

5.2.2. Deployment

To implement our framework, we adopted the implementation provided by [12] and implemented the classification and policy generation algorithms, which were not available in that framework. We employed Scikit-learn to implement the naive Bayes classifier. We normalized the dataset for the classification and split the dataset for training and testing into 80% and 20%, respectively, and then executed Algorithm 1. To implement Algorithm 2, we used python. For the algorithm, which needed a transition matrix to generate the policy, we first normalized the second dataset and used probabilistic analysis to generate the operator values for each context. This way, the algorithm would be able to generate the policy properly. We used Raspberry Pi 3 Model B as our IoT device and the details regarding the hardware and software are given in Table 4. We used Raspberry Pi 3 Model B as our IoT device and a desktop PC for the services. Raspberry Pi hosted the context broker, model generator, ABE policy generator, and the policy store. The desktop PC hosted the service provider, data store, decryptor, and KMS. Communication between Raspberry Pi and the desktop PC was achieved using a python websocket. The details regarding the hardware and software are given is Table 4.

Table 4. Hardware and Software Specification.

	Raspberry Pi	**Desktop PC**
Processor	1.2 GHz 64 bit quad-core ARM Cortex-A53	3.2 GHz 4 core
RAM	1 GB	8 GB
Storage	16 GB eMMC flash storage	320 GB
Operating System	Raspberian Debian OS	Ubuntu

In our experimentation, we used a python script to determine the resource utilization. The latency was measured using the difference between the time when the data were collected and when the encryption was completed. The decryption latency includes the decryption time along with the communication time to ask the context broker whether it has permissions to access the data or not.

5.3. Evaluation

The main objective of this experiments was to study the feasibility of our framework as well as the resource utilization. We have compared our algorithms with other ML algorithms, such as logistic regression, decision trees, random forest, and LSTM.

In order to evaluate the performance of our algorithm, we performed various experiments. For the access control evaluation, we experimented with logistic regression, decision trees, and naive Bayes. For the policy generator evaluation, we experimented with random forest, LSTM, and Markov chains. We analyzed the precision, recall, and F1-score of the algorithm, using the equations provided in [39].

We evaluated the algorithms using our dataset to determine the accuracy, precision, and recall, as well as the training time required to prepare the models, which are shown in Table 5. For experimentation and evaluation of the algorithm, we split 80% for training the models and 20% for testing the algorithm performances. As shown in Table 5 for the access control, the F-score of the logistic regression is higher, but the precision and training are lower. The decision tree has the lowest training time, but the naive Bayes gives the best results in terms of precision, F-score, and training time. For the policy generation evaluation, the lowest performance is provided by the random forest, but the training time is the lowest. LSTM has an average accuracy, but the training time is the highest among the other algorithms. Markov chain is the model in terms of all metrics. To summarize, from Table 5, we see that the accuracy of our algorithms is above 90% and their precision is above 85%.

Table 5. Evaluation of the Algorithms.

	Algorithm	Precision	Recall	F1	Training Time
Access Control	Logistic Regression	0.9183	0.8681	0.8925	0.0086
	Decision Tree	0.9248	0.8348	0.8755	0.0024
	Algorithm 1 (Naive Bayes)	0.9515	0.9062	0.9283	0.0037
Policy Generator	Random Forrest	0.8426	0.6775	0.7324	0.4581
	LSTM	0.8936	0.7976	0.8429	34.7515
	Algorithm 2	0.8926	0.913	0.9027	2.8564

Figures 7–9 are the experiments performed to find the resource utilization in our framework. In these experiments, we increased the policy length gradually with increments of five attributes, up to fifty attributes. The policy length is the total number of attributes in the policy. In Figure 7 is the CPU consumption with varying policy lengths. The CPU consumption is for encryption gradually increases as the length of the policy increases. On the other hand, decryption has almost the same CPU utilization when the number of policy attributes varies between 30 and 50. The CPU utilization does not go above 50% in the worst case, which means that the length of the policy compromises 50 attributes. Figure 8 is the memory utilization for encryption and decryption in megabytes. The memory utilization for encryption is lower than the memory utilization. Another interesting observation is that the memory utilization for decryption stabilizes after 35 attributes and is 25 attributes for the encryption. Lastly, Figure 9 is the overall time required for the encryption after data are received by the context broker. The execution time for decrypting is higher as it has to communicate with the context broker to obtain the permission for the data from the data store before it can start the process of decryption. From this figure, we see that, as the policy length increases, the execution time for the whole process increases from 0.5 s up to 16 s for encryption and 18 s for decryption.

In summary, the approach we proposed for developing our context-aware, adaptable, intelligent and lightweight security model consists of two main axes, namely, the dynamic creation of context-aware policies and the context-awareness dynamic encryption model. The conducted in-depth experiments proved the efficiency of both. Our framework ensures that the accuracy of our algorithms is above 90%, and their precision is around 85%, which is considerably high compared to the other approaches studied. However, many industrial aspects are still challenging and could affect our model accuracy rates, such as the complex identity management of multiuser, multiprofile, and multidevice technologies.

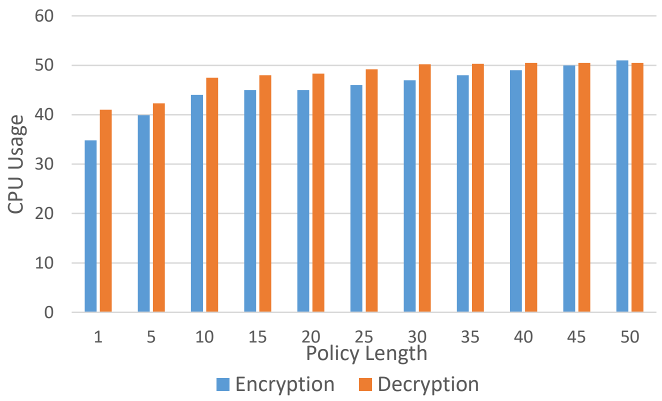

Figure 7. CPU utilization.

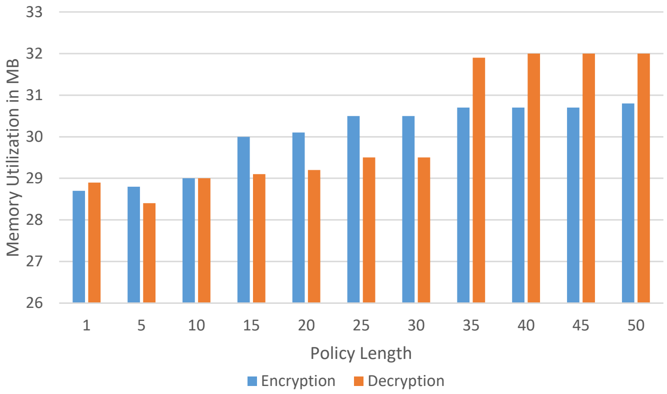

Figure 8. Memory utilization.

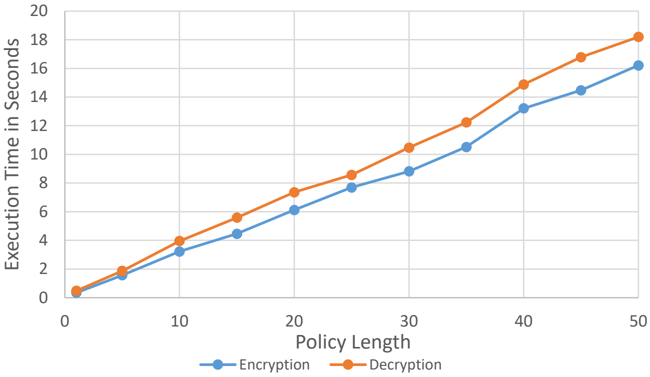

Figure 9. Execution time.

Furthermore, the implemented lightweight security framework is capable of granting secure interapplication data communication by encrypting all the requested sensitive sensor data and making them available for only the authorized applications according to the generated context-aware policies. Thorough experiments have been performed demonstrating the efficiency of CPU and memory utilization, as well as the execution time for the whole encryption/decryption process. More evaluation criteria might be considered for future improvements.

Finally, we believe that our smart adaptive model will open up promising directions for research and improvements using different ML techniques in heterogeneous IoT environments.

6. Conclusions and Future Work

In this paper, we have extended our previous work on LCA-ABE to provide a smart, adaptive, context-aware security model by adopting a novel intelligent dynamic creation of context-aware policies, which was achieved through smart-learning techniques. By leveraging attribute-based encryption (ABE), we also provided a context-awareness dynamic encryption model. Furthermore, our implemented security framework is capable of granting secure interapplication data communication by encrypting all of the requested sensor's sensitive data and making it available only for the authorized applications according to the predefined context-aware policies. Thorough experiments and evaluations, and by leveraging the full power of modern IoT smart environments and the newly imposed privacy regulations, we demonstrate the efficiency of the proposed solution.

We believe that our secure adaptive model will open up promising directions and can be used in many potential research applications for both academia and industry professionals, where context awareness, security, privacy, and access control are required. In the near future, we are planning to improve our model, using different ML techniques to solve problems in several military systems, such as command, control, communications, intelligence, and surveillance in different challenging and heterogeneous networking environments.

Author Contributions: Conceptualization, S.I., R.C. and C.T.; methodology, S.I. and R.C.; software, R.C.; validation, R.C. and H.O.-S.; formal analysis, S.I., R.C. and H.O.-S.; investigation, S.I., R.C. and H.O.-S.; resources, S.I. and R.C.; data curation, S.I. and R.C.; writing—original draft preparation, S.I. and R.C.; writing—review and editing, C.T., S.I., R.C. and H.O.-S.; supervision, C.T.; project administration, C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The authors are not making the dataset or the codes public due to privacy issue.

Conflicts of Interest: The authors certify that they have no conflicts of interest with or involvement in any organization or entity with any financial interest or non-financial interest in the the subject matter or materials discussed in this manuscript.

References

1. Schilit, B.N.; Theimer, M.M. Disseminating active map information to mobile hosts. *IEEE Netw.* **1994**, *8*, 22–32. [CrossRef]
2. Georgakopoulos, D.; Zaslavsky, A.; Perera, C. Sensing as a service and big data. In Proceedings of the International Conference on Advances in Cloud Computing (ACC'12), Bangalore, India, 26–28 July 2012. [CrossRef]
3. Text, O.L. General Data Protection Regulation (GDPR). 2022. Available online: https://gdpr-info.eu/ (accessed on 18 October 2022).
4. Ftc Privacy and Data Security. 2012. Available online: https://www.ftc.gov/policy/reports (accessed on 18 October 2022).
5. Cybersecurity and Infrastructure Security Agency CISA. 2022. Available online: https://www.cisa.gov/federal-information-security-modernization-act (accessed on 18 October 2022).
6. Sarker, I.H.; Colman, A.; Han, J.; Khan, A.I.; Abushark, Y.B.; Salah, K. Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model. *Mob. Netw. Appl.* **2020**, *25*, 1151–1161. [CrossRef]
7. Nawrocki, P.; Sniezynski, B.; Kolodziej, J.; Szynkiewicz, P. Adaptive context-aware service optimization in mobile cloud computing accounting for security aspects. *Concurr. Comput. Pract. Exp.* **2020**, *33*, e6070. [CrossRef]
8. Mshali, H.; Lemlouma, T.; Magoni, D. Adaptive monitoring system for e-health smart homes. *Pervasive Mob. Comput.* **2018**, *43*, 1–19. [CrossRef]
9. Alkhresheh, A.; Elgazzar, K.; Hassanein, H.S. Context-aware automatic access policy specification for iot environments. In Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 793–799. [CrossRef]
10. Sarker, I.H.; Salah, K. Appspred: Predicting context-aware smartphone apps using random forest learning. *Internet Things* **2019**, *8*, 100106. [CrossRef]
11. Phung, P.H.; Mohanty, A.; Rachapalli, R.; Sridhar, M. Hybridguard: A principal-based permission and fine-grained policy enforcement framework for web-based mobile applications. In Proceedings of the 2017 IEEE Security and Privacy Workshops (SPW), San Jose, CA, USA, 25 May 2017; pp. 147–156. [CrossRef]
12. Inshi, S.; Chowdhury, R.; Elarbi, M.; Ould-Slimane, H.; Talhi, C. LCA-ABE: Lightweight context-aware encryption for android applications. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; pp. 1–6. [CrossRef]
13. Selvan, S.; Mahinderjit Singh, M. Adaptive Contextual Risk-Based Model to Tackle Confidentiality-Based Attacks in Fog-IoT Paradigm. *Computers* **2022**, *11*, 16. [CrossRef]
14. Kim, J.C.; Chung, K. Neural-network based adaptive context prediction model for ambient intelligence. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1451–1458. [CrossRef]
15. Michalakis, K.; Caridakis, G. Context awareness in cultural heritage applications: A survey. *ACM J. Comput. Cult. Herit. (JOCCH)* **2022**, *15*, 1–31. [CrossRef]
16. Kavitha, D.; Ravikumar, S. IOT and context-aware learning-based optimal neural network model for real-time health monitoring. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4132. [CrossRef]
17. Jancic, A.; Warren, M.J. PKI-Advantages and Obstacles. In Proceedings of the AISM, Perth, Australia, 26 November 2004; pp. 104–114.
18. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98. [CrossRef]
19. Abowd, G.D.; Dey, A.K.; Brown, P.J.; Davies, N.; Smith, M.; Steggles, P. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 304–307. [CrossRef]

20. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 414–454. [CrossRef]
21. Sarker, I.H.; Kayes, A.; Watters, P. Effectiveness analysis of machine-learning classification models for predicting personalized context-aware smartphone usage. *J. Big Data* **2019**, *6*, 57. [CrossRef]
22. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334. [CrossRef]
23. Zhang, Y.; Deng, R.H.; Xu, S.; Sun, J.; Li, Q.; Zheng, D. Attribute-based encryption for cloud computing access control: A survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–41. [CrossRef]
24. Oberko, P.S.K.; Obeng, V.H.K.S.; Xiong, H. A survey on multiauthority and decentralized attribute-based encryption. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *13*, 515–533. [CrossRef]
25. Ullah, A.; Azeem, M.; Ashraf, H.; Alaboudi, A.A.; Humayun, M.; Jhanjhi, N. Secure healthcare data aggregation and transmission in IoT—A survey. *IEEE Access* **2021**, *9*, 16849–16865. [CrossRef]
26. Ambrosin, M.; Conti, M.; Dargahi, T. On the feasibility of attribute-based encryption on smartphone devices. In Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, Florence, Italy, 21 May 2015; pp. 49–54. [CrossRef]
27. Maheswari, S.; Gudla, U. Secure sharing of personal health records in Jelastic cloud by attribute based encryption. In Proceedings of the 2017 fourth International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–4. [CrossRef]
28. Akinyele, J.A.; Pagano, M.W.; Green, M.D.; Lehmann, C.U.; Peterson, Z.N.; Rubin, A.D. Securing electronic medical records using attribute-based encryption on mobile devices. In Proceedings of the first ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, Chicago, IL, USA, 17 October 2011; pp. 75–86. [CrossRef]
29. Taha, M.B.; Chowdhury, R. GALB: Load Balancing Algorithm for CP-ABE Encryption Tasks in E-Health Environment. In Proceedings of the 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Bangalore, India, 26–27 November 2020; pp. 165–170. [CrossRef]
30. Chowdhury, R.; Ould-Slimane, H.; Talhi, C.; Cheriet, M. Attribute-based encryption for preserving smart home data privacy. In *International Conference on Smart Homes and Health Telematics*; Springer: Cham, Switzerland, 2017; pp. 185–197. [CrossRef]
31. Tan, Y.L.; Goi, B.M.; Komiya, R.; Tan, S.Y. A study of attribute-based encryption for body sensor networks. In *International Conference on Informatics Engineering and Information Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 238–247. [CrossRef]
32. Shao, J.; Lu, R.; Lin, X. Fine-grained data sharing in cloud computing for mobile devices. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2677–2685. [CrossRef]
33. Taha, M.B.; Ould-Slimane, H.; Talhi, C. Smart offloading technique for CP-ABE encryption schemes in constrained devices. *SN Appl. Sci.* **2020**, *2*, 274. [CrossRef]
34. Inshi, S.; Elarbi, M.; Chowdhury, R.; Ould-Slimane, H.; Talhi, C. CAPEF: Context-Aware Policy Enforcement Framework for Android Applications. *J. Eng. Res. Sci.* **2023**, *2*, 13–23. [CrossRef]
35. Annane, B.; Alti, A.; Lakehal, A. Blockchain based context-aware CP-ABE schema for Internet of Medical Things security. *Array* **2022**, *14*, 100150. [CrossRef]
36. Annane, B.; Alti, A.; Laouamer, L.; Reffad, H. Cx-CP-ABE: Context-aware attribute-based access control schema and blockchain technology to ensure scalable and efficient health data privacy. *Secur. Priv.* **2022**, *5*, e249. [CrossRef]
37. Raftery, A.E. A model for high-order Markov chains. *J. R. Stat. Soc. Ser. (Methodol.)* **1985**, *47*, 528–539. [CrossRef]
38. Ching, W.K.; Huang, X.; Ng, M.K.; Siu, T.K. Higher-order markov chains. In *Markov Chains*; Springer: Boston, MA, USA 2013; pp. 141–176. [CrossRef]
39. Dalianis, H., Evaluation Metrics and Evaluation. In *Clinical Text Mining: Secondary Use of Electronic Patient Records*; Springer International Publishing: Cham, Switzerland, 2018; pp. 45–53. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.