Expert Systems With Applications 213 (2023) 119271

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

AC-ABAC: Attribute-based access control for electronic medical records during acute care☆

Marcela T. de Oliveira [a,b,*], Yiannis Verginadis [c,d], Lúcio H.A. Reis [a], Evgenia Psarra [d], Ioannis Patiniotakis [d], Sílvia D. Olabarriaga [a]

[a] Department of Epidemiology and Data Science, Amsterdam University Medical Centers, University of Amsterdam, Amsterdam, Netherlands
[b] Department of Biomedical Engineering and Physics, Amsterdam University Medical Centers, University of Amsterdam, Amsterdam, Netherlands
[c] Athens University of Economics and Business, Athens, Greece
[d] Institute of Communications and Computer Systems, Athens, Greece

ARTICLE INFO

Keywords:
Attribute-based access control
XACML
Electronic Medical Records
Acute care
Data privacy
Cloud storage

ABSTRACT

Acute care demands fast response and procedures from the healthcare professionals involved in the emergency. The availability of electronic medical records (EMR) enables acute care teams to access patient data promptly, which is critical for proper treatment. The EMR contains sensitive data, so proper access control is a necessity. However, acute care situations entail the introduction of dynamic authorisation techniques that are able to swiftly grant access to the acute care teams during the treatment and that at the same time can revoke it as soon as the treatment is over. In this work, our contributions are threefold. First, we propose a step-by-step methodology that defines dynamic and fine-grained access control in acute care incidents. Then, we applied this methodology with the Amsterdam University Medical Center acute stroke care teams, resulting in a new model coined 'Acute Care Attribute-Based Access Control (AC-ABAC)'. AC-ABAC implements access control policies that take into account contextual attributes for dynamically sharing patient data with the appropriate healthcare professionals during the life cycle of acute care. Finally, we evaluate the performance and show the feasibility and correctness of AC-ABAC through a prototype implementation of the model and simulation of patient data requests in various scenarios. The results show that the most complex policy evaluation takes on average 194.89 ms, which is considered worthwhile when compared to the added value to the system's security and the acute care process.

1. Introduction

The protection of privacy of a patient's Electronic Medical Record (EMR) is imperative, even in emergencies. In critical access control systems, static control rules are typically applied, which usually involve the role (e.g., doctor) or even an explicit enumeration of individuals that should be allowed to access a patient's EMR. As a result, in emergencies, we witness the so-called 'break-glass' procedure (de Oliveira et al., 2020), during which medical personnel may bypass rigid access control rules and acquire access to a patient's medical history. However, we advocate that access control under emergency conditions should be supported with the required dynamism instead of adopting a break-glass procedure. Furthermore, in many cases, parts of the patient's EMR remain unreachable even in emergencies because they are located in information systems outside the treating hospital's boundaries. Therefore, the availability of EMR across organisational boundaries has been proposed to enhance the quality of information available for decision-making during acute care (Hillestad et al., 2005).

Cloud storage services match the needs of remote and ubiquitous access to medical data for multiple healthcare organisations. However, security and privacy challenges still hamper the wide adoption of cloud services. Moreover, patients and healthcare organisations are afraid of losing control over the EMR when storing it on untrusted third-party clouds (Abbas & Khan, 2014). In May 2018, the General Data Protection Regulations (GDPR) (European Union) came to reinforce the need for personal data protection, defining conditions for data sharing and processing across multiple domains. Under the GDPR, healthcare

☆ MTO and YV contributed equally to the paper.

* Corresponding author at: Department of Epidemiology and Data Science, Amsterdam University Medical Centers, University of Amsterdam, Amsterdam, Netherlands.

E-mail addresses: m.tuler@amsterdamumc.nl, m.tuler@amasterdamumc.nl (M.T. de Oliveira), jverg@mail.ntua.gr (Y. Verginadis), l.henrikamorimreis@amsterdamumc.nl (L.H.A. Reis), jennypsarraemp@mail.ntua.gr (E. Psarra), ipatini@mail.ntua.gr (I. Patiniotakis), s.d.olabarriaga@amsterdamumc.nl (S.D. Olabarriaga).

https://doi.org/10.1016/j.eswa.2022.119271
Received 17 May 2021; Received in revised form 29 September 2022; Accepted 12 November 2022
Available online 18 November 2022
0957-4174/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

organisations, the 'data controllers', have accountability for fulfilling these regulatory requirements. The accountability relies on their ability to demonstrate that appropriate procedural security measures are being applied and, most importantly, compliant with the GDPR. When a single cloud-based EMR system is used, the GDPR classifies healthcare organisations as joint data controllers. These jointly determine 'why' and 'how' personal data should be processed for complying with the GDPR rules designed specifically for healthcare data processing (European Commission). Therefore, a cloud-based EMR system's access control mechanisms should be designed to support multi-organisation collaboration and offer accountability and auditability at individual, team and organisation levels.

The main goal of an EMR system is patient data availability; therefore, the access control must not block any rightful request for the sake of the patients' vital interest. Because of that, the access control models usually are more permissive than needed for patient treatment. This may pose threats to patient privacy (Bhuyan et al., 2020) because the users might abuse the permissions and use the data for other purposes than treating a particular patient, for example, for curiosity sake. Researchers have proposed using the Attribute-Based Access Control (ABAC) model to achieve a more fine-grained access control; however, its adoption in real healthcare applications remains challenging. One reason is that the information workflow during acute care involves cross-organisation data sharing, which is complex and difficult to understand and model adequately. Consequently, the existing access control models using ABAC usually cover well only the conventional access situations (e.g. doctor appointments), leaving the acute care case less protected.

We consider that understanding the information workflow during treatment is fundamental for adequate access control modelling to protect patient privacy without compromising legitimate data availability. Therefore, one of the innovations in this paper is the methodology proposed to apply the Context-Aware ABAC model in the acute care case, which leverages a fine-grained access control modelling for EMR systems. This step-by-step methodology guides understanding who the subjects are, their actions, and the resources to be protected. Then, based on the context, the developer can propose rules combining contextual attributes belonging to the subject, resource or the environment. The developer must analyse all the rules to guarantee that the information needed to evaluate the rules is available. Finally, the rules can be combined as access control policies and enforced into an EMR system.

The second innovation concerns the application case, which covers a difficult access control situation in acute care. Our major contribution is to propose an access control model that will keep the patient's EMR confidential and private without compromising the data availability for the legit acute care teams. Our proposal applied the step-by-step methodology with the Amsterdam UMC acute care teams. By means of a thorough analysis guided by the methodology, we understood which rules and contextual attributes should and should not be used in the modelling. The resulting model, coined Acute Care Attribute-Based Access Control (AC-ABAC), presents the policies and contextual attributes used by acute care teams to legitimate the emergency session for a patient. The emergency session dynamically includes the teams and professionals involved in the patient treatment according to the demand, granting and revoking access to the patient's data along the treatment timeline and workflow. Moreover, we developed the AC-ABAC prototype to demonstrate the feasibility of our approach as well as to evaluate the correctness and performance of the model. All the prototype code is publicly available.

In summary, our work addresses the challenges of integrating ABAC modelling with an EMR system with the following contributions:

- We propose a step-by-step methodology to define dynamic and fine-grained access control to patient data using Context-Aware Attribute-Based Access Control.
- We present the Acute Care Attribute-Based Access Control model to keep patient data confidential and private without compromising data availability for the legit acute care teams.
- We present a prototype including a Context-Aware Attribute-Based Access Control system and the AC-ABAC model implementation. The code and the simulation results obtained with the prototype are available at de Oliveira, Verginadis, Reis, Evgenia Psarra, and Olabarriaga (2021a), which can be used to reproduce, verify and expand our research.

2. Related work

An essential milestone in the access control field is the usage of roles (Role-based Access Control, RBAC). RBAC is an approach that separates users and their permissions regarding resources and places the data requester role at the centre (Mitra, Sural, Vaidya, & Atluri, 2016; Rao, Nayak, Ray, Rahulamathavan, & Rajarajan, 2021). Various EMR systems consider that the decision to uncover patient data is affected by various factors that comprise the patient's situation, such as a regular medical appointment or an emergency treatment (Arora & Gosain, 2020; Ferreira et al., 2009; Nazerian, Motameni, & Nematzadeh, 2019). Indeed, one of these factors is the healthcare professional role of who requests the data, but it is not the only one.

Nazerian et al. (2019) proposed Emergency RBAC (E-RBAC), which defines emergency roles based on the user trust level and gives access permission to patient data for those who have an emergency role. Arora and Gosain (2020) proposed a detector to analyse the misuse of E-RBAC through analyses of audit logs to classify users' trust levels. Our proposal also considers that trust is given to all professionals in every healthcare organisation.

In 2008, Peleg, Beimel, Dori, and Denekamp (2008) highlighted the problems with the RBAC model used in the existing EMR systems and proposed a situation-based access control approach based on scenarios of data requests. In addition, the authors proposed a generic method of interviewing healthcare stakeholders and understanding their data needs. Our work focused on the acute care situation and proposed a structured methodology to identify the data needs and legitimate access during the emergency.

Lunardelli, Matteucci, Mori, and Petrocchi (2013) proposed an analytic hierarchical process model for solving policy conflict issues in EMR systems. They created a prototype and analysed the system performance which was using the eXtensible Access Control Markup Language (XACML). Calvillo-Arbizu, Román-Martínez, and Roa-Romero (2014) proposed an access control mechanism based on XACML and ABAC, which conforms to ISO 13606. Furthermore, the proposed system applies an ontology for automatic reasoning to the authorisation process. In our previous work (Verginadis et al., 2017), we worked on an XACML-based access control system for authorising access to sensitive data persisted in cloud resources. That work was extended in this paper to enable further the dynamism of policies for usage in an acute healthcare situation. Moreover, in Psarra, Verginadis, Patiniotakis, Apostolou, and Mentzas (2020a), we have introduced a model of concepts and properties called ASCLEPIOS Context-Aware Security Model (CASM). This model serves as the background ontology required for creating access control policies for EMRs in acute care conditions.

Abomhara, Yang, and Køien (2016) proposed a work-based access control model that modifies the user-role assignment model through the concept of team role assigned to a treatment. Seol, Kim, Lee, Seo, and Baik (2018) propose a cloud-based EMR model that performs attribute-based access control using XACML. The authors mentioned the possibility of emergencies and assumed that the system would decide based on this information. However, Abomhara et al. (2016) and Seol et al. (2018) do not specify how to legitimate teams and professionals during emergency treatment.

Inspired by the related work, we propose a methodology to identify the contextual attributes that legitimate the emergency. These

Table 1
Comparison of relevant state-of-the-art works.

Research works	Non-ABAC	ABAC	XACML	Methodology	EMR System	Acute care
Mitra et al. (2016)	✓					
Rao et al. (2021)	✓					
Ferreira et al. (2009)	✓					✓
Nazerian et al. (2019)	✓				✓	✓
Arora and Gosain (2020)	✓					✓
Peleg et al. (2008)	✓			✓	✓	
Lunardelli et al. (2013)		✓		±[a]	✓	✓
Calvillo-Arbizu et al. (2014)	✓		✓		✓	
Verginadis et al. (2017)		✓	✓			
Psarra et al. (2020a)		✓			✓	
Abomhara et al. (2016)	✓				✓	
Seol et al. (2018)		✓	✓		✓	±[b]
(this work)		✓	✓	✓	✓	✓

[a]Methodology for conflict resolution.
[b]Partially support acute care.

attributes can be used to dynamically grant and revoke access to patient EMR for the acute care teams involved in the emergency session. Furthermore, we develop a model to aggregate these attributes to the ABAC engine and use them to evaluate rules and policies. In Table 1, we position the contributions of our work in comparison to the related works mentioned in this section. Specifically, the ABAC or non-ABAC capabilities of these works are listed in the first columns. Notice that most of these related works that lack the ABAC models implement the RBAC paradigm. The second column depicts the reference implementation of the XACML standard, while the third column lists the related works that introduce or adopt a methodology that may lead to appropriate policies' modelling. The fourth column refers to whereas the integration into an EMR system is realised. Finally, the last column depicts the consideration of acute care situations in the enforced access controls. The axes of this comparison are mainly justified by the significant advantages that ABAC approaches exhibit in contrast to other traditional access control paradigms, as it has been argued before (Verginadis et al., 2017). To the best of our knowledge, no previous work has proposed a dynamic access control methodology that resulted in an access control model for cross-organisation data sharing in acute care scenarios.

3. Stakeholders of electronic medical records during acute care

This section introduces stakeholders of an acute care EMR system and their roles in the ABAC paradigm. We specifically consider the situation when a patient is treated in an Emergency Session (ES), covering the time window since the patient requests emergency treatment until discharge.

We describe the acute stroke care case involving professionals from the emergency call centre, ambulance services and hospitals. The professionals with different roles are organised in teams in each organisation. The time interval in which the teams participate in the patient's ES is the team's Episode of Care (EC), according to FHIR standard concept (FHIR). An EC starts when a team is invited to the ES and ends when a team finishes the treatment. After that, access to the data is revoked. During the EC, the team members can read and update the patient's EMR.

We identified three recurrent scenarios in which the stakeholders interact with the patient's EMR during acute care: In the first, the patient or someone on behalf of the patient contacts the call centre and requests an ambulance that takes the patient to the hospital. In the second scenario, the patient or someone on behalf of the patient contacts the call centre, which informs which hospital the patient needs to be taken to by their own transportation means. In the third scenario, the patient goes directly to the nearest hospital, where the ES is started and can be extended to other teams if necessary. Fig. 1 illustrates the three scenarios with episodes of care by the call centre, ambulance and hospital teams. For each team, the figure presents the starting and ending times of their involvement in the ES. It also shows when a team invites another team to join the ES. In all situations, the teams will have access to the patient's EMR from the moment of their involvement in the ES until their task in the treatment is completed. Below, we describe the EC for the call centre, ambulance, and hospital teams.

Call centre team. The phone call event is the beginning of the patient's ES in the first and second scenarios. An emergency call centre professional receives the call from the patient or someone on behalf of the patient. During the phone call, the professional follows a triage protocol and needs to read the patient's EMR and adds new information about the patient's current condition. In the first scenario, the professional decides to request an ambulance team to pick up the patient. The ambulance team that accepted the request then also becomes involved in the patient's ES. Suppose now that the professional decides to request an ambulance acute care team to pick up the patient. The ambulance team that accepted the request then also becomes involved in the patient's ES. In another scenario, the patient might use private transportation, so the acute care team designated to treat the patient at the hospital also takes part in the patient's ES. In both cases, as soon as the patient is under treatment by one of the acute care teams, the call centre professional leaves the ES and should no longer have access to the patient's EMR.

Ambulance team. Ambulance acute care team professionals must have access to a patient's EMR from the emergency request until the patient's delivery at the hospital. First, the ambulance professionals notify the EMR system that the patient was picked up. Then, following the triage, the ambulance team requests an adequate hospital to receive the patient. After the request and acceptance by the hospital, the ambulance starts the transportation. Finally, after delivering the patient to the hospital, the ambulance professionals have extra time to complete data into the patient's EMR.

Hospital team. As soon as the hospital team is involved in the patient's ES, its members should read the patient's EMR to better prepare for the treatment. The hospital is requested by an ambulance team to receive the patient or receives a patient that comes directly to the hospital using private transportation. During the treatment, the acute care team, can add new records to the patient's EMR. In the case of transfer to another hospital, a second ambulance and hospital teams become involved in the ES and access the patient's EMR. The ES and the ability to read the patient's EMR terminate when the patient is transferred or discharged. However, the team members should have extra time to complete the treatment record after the ES is over.

Note that each team member's data processing actions must be recorded in the audit logs at the user level, as this creates full responsibility for the user and his actions undertaken during an emergency.

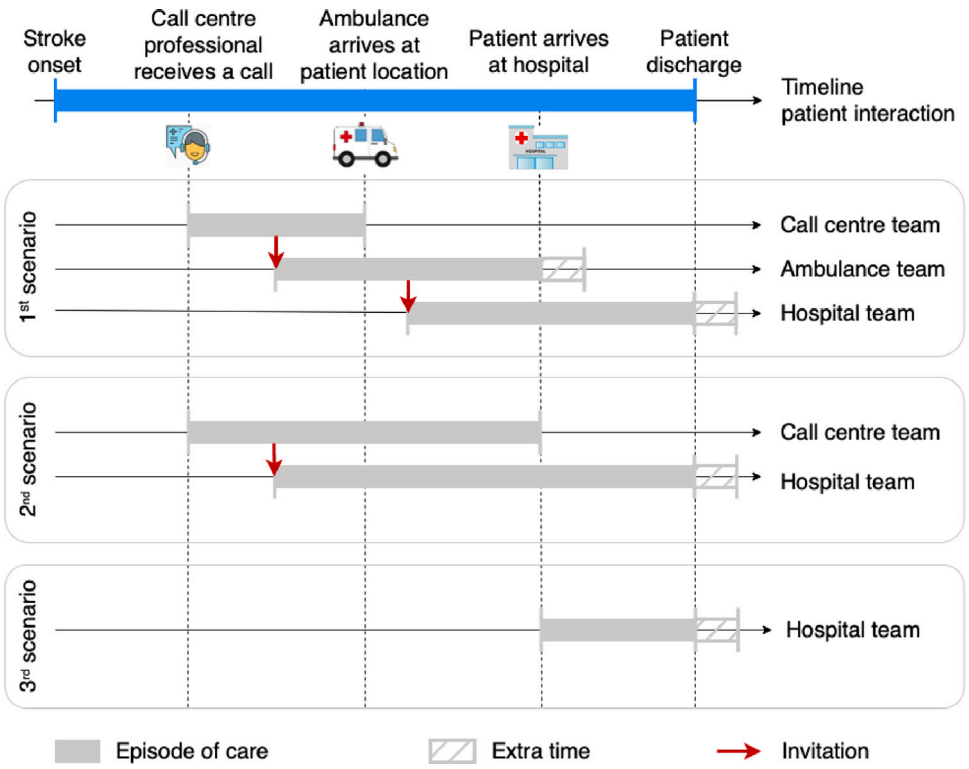

Fig. 1. Timeline of the teams' involvement during acute care from the stroke onset until patient discharge in the three scenarios. Each team is involved during the periods marked in grey, corresponding to the Episode of Care, during which the team members have access to the patient's medical record. The hatched marks indicate extra time given for completing the report in the medical record after the involvement with the patient ends.

4. Attribute-based access control architecture for electronic medical records

ABAC defines an access control paradigm in which access rights are granted to the requester using policies consisting of logical combinations of contextual attributes. Fig. 2 presents the main architecture entities and their direct communication flow following the ABAC model's reference implementation using the eXtensible Access Control Markup Language (OASIS, 2020a). XACML is an OASIS (OASIS, 2020b) standard that describes both a policy language and an access control decision request/response language. Both languages use XSD (XSD) notations; hence, policy definition and request/response elements are serialised as XML elements. The standard defines five main components that handle access decisions, namely, Policy Enforcement Point (PEP), Policy Administration Point (PAP), Policy Decision Point (PDP), Policy Information Point (PIP), and a Context Handler. In our previous work, we extended this reference implementation by providing the appropriate integration hooks to external systems to facilitate integration. We also presented a policy editing component named ASCLEPIOS Models and PoLicies Editor (AMPLE) (Psarra, Verginadis, Patiniotakis, Apostolou, & Mentzas, 2020b) for managing policies and multiple context handlers of different complexity.

Our system includes the architectural components depicted in Fig. 2 and briefly described below.

Electronic Medical Records (EMR) system is a web system responsible for the management and storing of the encrypted EMR and respective cryptography keys. Moreover, the EMR system offers the data persistence layer and the services to process data, e.g. create, read, update and delete (CRUD), controlled through PEPs.

Subjects refer to any entity that can interact with the EMR system to request data access. A subject has one or more attributes for characterisation in the system. We consider two different types of subjects: patients and healthcare professionals.

Resource is a data, service or system component that needs to be protected through access control. Each resource offers specific actions that require data processing. A request action refers to the subjects' intended action (e.g. read or update) over a specific resource. For example, the patient data in the EMR system are represented in encrypted form as resources protected through ABAC. Also, the respective cryptography keys could be managed by the EMR system and protected with ABAC.

Environment elements provide contextual information about the requester or resource. This information can come from the access requester, such as the timestamp, IP address and geolocation, or from external smart devices.

A *Requester* embodies the user application. The subject sends access requests through the user application to process any resource of the EMR system.

The *AMPLE editor* is a graphical web tool that allows the data controller to create, persist and update XACML-based access control policies. These policies are context-aware because they imply evaluating the contextual attributes of subjects, resources, actions and the environment before yielding a permit or deny decision.

The *Policy Administration Point (PAP)* stores a database used for persisting policies and access request decisions. The PAP provides access to the pool of defined policies that have been deployed and activated through the AMPLE editor. Nevertheless, these policies may have static or dynamic parameters that can be updated even at run-time to be immediately enforced.

The *Policy Enforcement Point (PEP)* constitutes the integration hook to any external system, such as an EMR system. It manages and serves incoming access requests for processing the patient's EMR. Different PEPs may be used in various sub-components of the EMR system, or they can be appended to the application server used (e.g., Tomcat). A PEP can receive access requests and freeze the execution workflow until a decision is yielded. At the same time, it propagates the requests and attributes to the ABAC system's decision-making components.

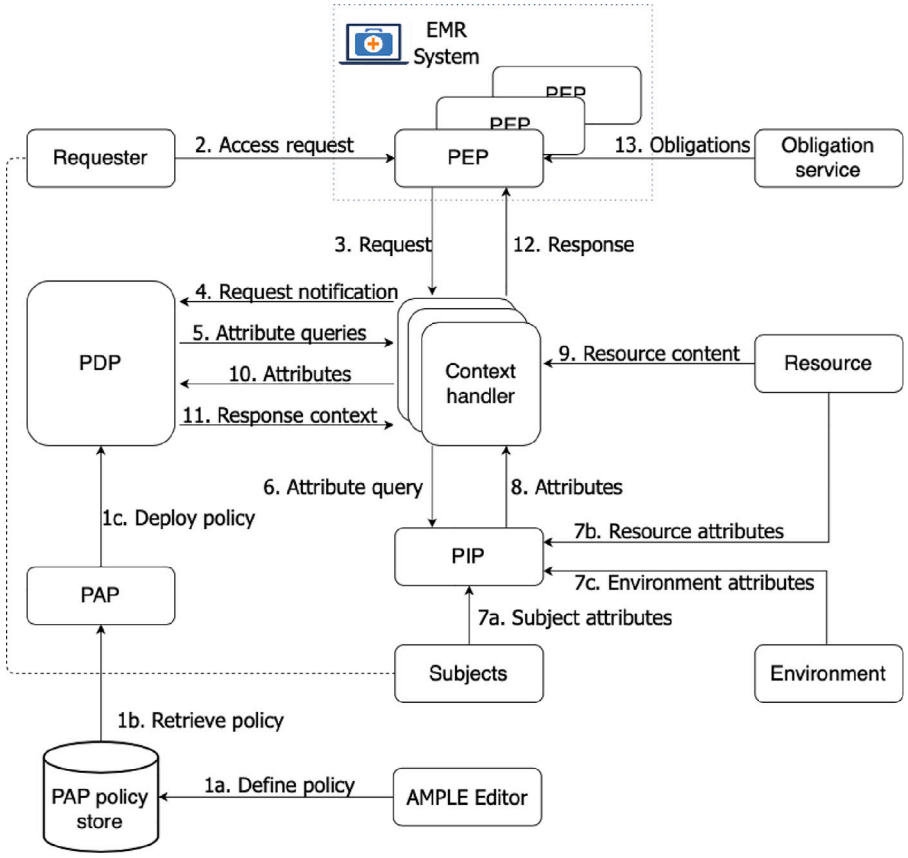

Fig. 2. Context-Aware Attribute-based Access Control System architecture and communication flow.

The *Policy Decision Point (PDP)* is the core decision place for any incoming access request intercepted by a dedicated PEP. It collects all the necessary contextual information and yields an access control decision, permit or deny, according to the defined policies.

The *Policy Information Point (PIP)* is responsible for retrieving the necessary attributes for the policy evaluation from several external or internal entities. The attributes aggregated to the PIP may be retrieved from the resource to be accessed (partial or complete EMR), the environment, subjects and the intended action. The attribute values refer mainly to raw information.

Context Handlers are responsible for semantically uplifting the raw information received by a PIP and producing the appropriate level of contextual information for the policies. Thus, they enable the aggregation of dynamic attributes to context-aware policies.

The *Obligation service* is a directive from the PDP to the PEP on what must be carried out before or after the access request is approved. If the PEP is unable to comply with the directive, even the approved access request must not be realised. The augmentation of obligations eliminates the gap between formal requirements and policy enforcement. An example of an obligation could be sending the "purpose of access" declaration for all types of access requests. If the PEP does not receive a valid value, the directive does not comply, and the access request is denied.

Note that the Context-Aware ABAC integrated with the EMR system can dynamically digest new policies at run-time. This means that, upon proper authentication, the PAP storage can be updated with new policies to be enforced, and the mapping of context handlers for inferring the context attribute values can be changed on the fly without interfering with the current policies in the system.

5. Methodology for dynamic and fine-grained access control model

This section proposes a methodology that leverages a fine-grained access control mechanism to the patient's EMR based on the ABAC paradigm. Fig. 3 depicts the methodology phases described below, namely *Preparation*, *Analysis*, *Development*, *Policies definition* and *Policies enforcement*.

In the *Preparation phase*, we prepare a template to register access control policies and the respective stakeholders for each use case scenario. The template can be found on supplemental material and involves a short description of the objectives and resources that must be protected. Moreover, it provides placeholders for expressing context-driven access control rules through its tabular format, i.e., to list the requester, action, resource, environment, logical operators that combine rules and the desired access control decision. During the interview with each stakeholder, we fill the template with all the relevant emergency procedures specified concerning the need to access the EMR. Thus, the template constitutes the base for extracting the appropriate contextual information that should bind the access control decisions (i.e., the ABAC policies).

The *Analysis phase* involves analysing the filled templates by investigating the required access control rules from the requester, the intended actions, and the resources to be accessed. The purpose is to enumerate the rules that must be used along with the contextual attributes considered per rule. Thus, a significant part of the analysis phase involves determining whether the contextual information needed for access control can be acquired or inferred from the EMR system. The selection of the appropriate contextual attributes is of critical importance for defining dynamic access control policies. In our previous

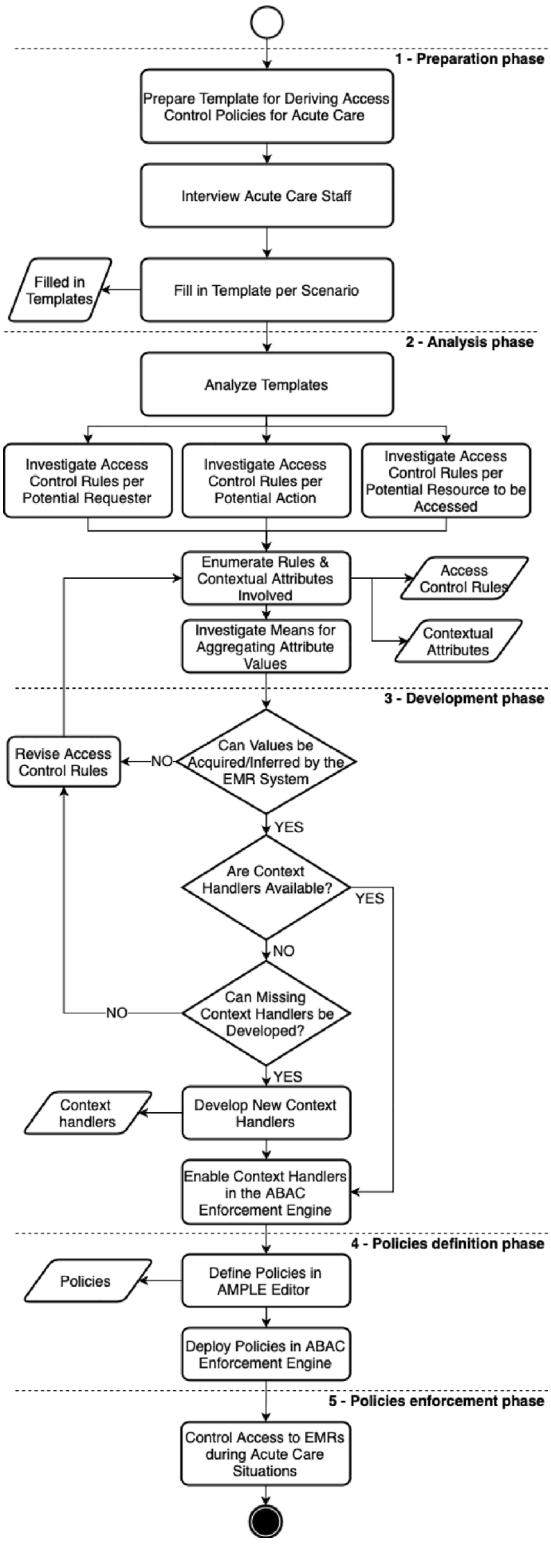

Fig. 3. Methodology for defining context-aware ABAC policies.

work, we have defined CASM (Psarra et al., 2020a), an ontology that serves as a basis for creating ABAC policies. That ontology is used here to map the contextual attributes involved in access control policies. For a specific policy to be enforced on an incoming access request, the system needs to acquire values for the contextual attributes involved, which happens through context handlers.

The *Development phase* refers to dedicated software (i.e., context handlers) that can leverage raw data to semantically enriched information. If the contextual attributes cannot be acquired or inferred, then the access control rules are revised. Any missing context handlers should be developed and enabled before using the corresponding context to access control policies.

During the *Policies definition phase*, the data controller of the EMR system defines the context-aware policies using the AMPLE policy editor. Through AMPLE, the policies are defined based on context-aware rules and then they are serialised in XACML, which is an appropriate format for enforcement.

The *Policies enforcement phase* is activated once the policies expressed in XACML are deployed in a dedicated ABAC engine. The ABAC enforcement engine retrieves all the related contextual attributes, infers the missing ones by invoking the relevant context handlers and yields a permit or deny decision according to the policies evaluation.

6. Attribute-based access control modelling for acute care

This section describes the Acute Care Attribute-Based Access Control (AC-ABAC) modelling resulted from applying the proposed methodology.

Furthermore, AC-ABAC follows the GDPR requirement for data confidentiality and privacy. According to Art.6 of the GDPR - "Lawfulness of processing: Processing shall be lawful only if and to the extent that at least one of the following applies: ...d) processing is necessary in order to protect the vital interests of the data subject or of another natural person". Therefore, Art. 6 imposes that data access during acute care must be granted for those professionals involved in the treatment, and only during the treatment, and then revoked when the treatment is over.

AC-ABAC protects any resource that makes the electronic medical records available for the acute care teams. For instance, the resource could be the patients' encrypted records and the respective cryptographic keys. We believe that a cryptography scheme combined with dynamic access control would provide medical systems with the confidentiality and data privacy required. The implementation of such encryption protocols is no trivial matter, which others have explained (Michalas, 2019). For simplicity, we kept the cryptographic part out of the scope of this paper so that we focus on the access control modelling.

AC-ABAC consists of policies and contextual attributes definitions for coping with dynamic and efficient access control needed in acute care situations. Note that policies considering professionals' roles, IP address, and secure connection are essential and have been explored in previous research (Psarra et al., 2020b). Our focus here is to have legitimate access to patient data during an emergency session for the acute care teams involved in the patient treatment. In addition, we consider the interaction between the professionals and teams from multiple organisations as an anchor of trust for access control modelling. The results are presented following each phase of the methodology (see Fig. 3).

6.1. Preparation phase

We interviewed professionals from the Amsterdam UMC hospital that work close to the call centre and ambulance service. The template presented in Section 5 was filled to collect information regarding *subjects*, *actions*, *resources*, and *contextual attributes* (see also de Oliveira, Verginadis, Reis, Evgenia Psarra, & Olabarriaga, 2021b). For members of the call centre, ambulance and hospital teams, we listed an entry in the template for reading and updating the patient EMR. Then, we determine the immutable and dynamic attributes related to each type of subject, action and resource, and the respective *expected outcome* (permit or deny).

Table 2
Contextual attributes, definitions and the subject of the attribute.

Contextual attribute	Definition	Belongs to
$Patient_{ID}$	Identification of patient under emergency treatment.	User
$User_{ID}$	Healthcare professional identification.	User
$Team_{ID}$	Team identification within an acute care team.	Team
$Team_{tag}$	Team type, where $tag \in [c, a, h]$.	Team
$Starter_{ID}$	Identification of healthcare professional who started ES_{ID}.	ES
ES_{ID}	Emergency session identification.	ES
$t_{startshift}$	Timestamp of when the professional starts the shift.	User
$t_{endshift}$	Timestamp of when the professional ends the shift.	User
$t_{request}$	Access request timestamp.	User
t_{invite}	Invitation timestamp in EC of the $Team_{ID}$ to attend a patient's ES.	Team
t_{treat}	Starting treatment timestamp in the EC of the $Team_{ID}$ in the ES.	Team
t_{revoke}	Revocation timestamp in the EC of the $Team_{ID}$ in the ES.	Team

6.2. Analysis phase

We observed that the *subjects* are the active healthcare professionals in the acute care teams involved in the patient's ES. Therefore, the *resource* should be the patient EMR, and the *actions* should be limited according to the involvement of teams during the treatment timeline. Healthcare professionals should be able to read the EMR as soon as they are involved in the ES. Still, they only should be able to write data on the EMR after they start treating the patient. Two exceptions are the call centre professional and the hospital, who interact with the patient by phone or directly in the emergency department, and can read and update data from the beginning of the call. Moreover, after the treatment is over, the professionals involved should have extra time to add new data about the recent EC. Every EC in the ES is limited by a timeout value that varies according to the acute care team type.

The combination of *contextual attributes* legitimates the patient's ES. These characterise the patient, the healthcare professionals, the acute care team involved in the ES and the duration time of each team's EC on the ES. Table 2 lists the contextual attributes that are dynamically assigned to professionals and acute care teams during the ES. $Starter_{ID}$ is the member of the first team who creates the ES_{ID} and associates it to the $Patient_{ID}$. Every team has a $Team_{tag}$, where *tag* characterises the different types of teams for call centre (c), ambulance (a) and hospital (h). Every team member can request access to read or update data on the patient EMR and invite another team to participate in the ES. However, only $Team_c$ and $Team_h$ can start an ES, only $Team_a$ and $Team_h$ can have extra time to update after revoke time, and only $Team_h$ can discharge patient.

Table 3 enumerates the rules and contextual attributes values involved per entity. The request for reading or updating data must contain the following attributes: $t_{request}$, requester $User_{ID}$, $Team_{ID}$ and $Patient_{ID}$. When a healthcare professional joins a team, it is created an entry in the TeamMembers table, which contains the $Team_{ID}$ and the $User_{ID}$ of the professional. When an ES is started, it creates an entry in the ES table with the $Patient_{ID}$ and the $Starter_{ID}$. When a team joins an ES, it is created an entry in the EC table with the $Team_{ID}$ that is responsible for the episode and the ES_{ID} that it belongs to. Moreover, each entry on the EC table contains the $t_{request}$, t_{invite}, t_{treat} and t_{revoke} attributes describing the team participation timeline in the ES. These attributes are evaluated according to these tables and the contextual attributes defined on Table 2.

6.3. Development phase

Following the steps in phase 2, Fig. 3, we investigated means to aggregate the contextual attribute values through context handlers. The context handlers must be able to dynamically either infer the attribute values from the request or acquire them from the environment. Following the methodology, we have developed the necessary context handlers to aggregate each contextual attribute from PIP since none were available. Note that user interactions with the EMR System aggregate the specific contextual attributes listed in Table 2. After an action is taken, the contextual attributes' values are created or updated in the PIP, often by the EMR System. Therefore, during policy evaluation, the context handlers acquire the contextual attribute values from the PIP.

The following actions trigger the changes in the PIP: When the healthcare professional starts and ends the work shift, it creates $t_{startshift}$ and $t_{endshift}$. When the organisation's administrators manage teams by adding or removing members, the PIP updates the teams, indexing with $Team_{ID}$ in the TeamMembers table. The ES attributes and the involved teams are updated when the teams act on the EMR System. For example, when $Starter_{ID}$ initiates the ES, it creates an entry on the ES table with ES_{ID}, and also associates the $Patient_{ID}$ and $Starter_{ID}$, so that all information about the ES becomes available on the PIP through the appropriate context handler. Every team that participates in the ES has an EC that starts with t_{invite}, has t_{treat} and ends with t_{revoke}. The 'previous' and 'next' teams are coined regarding the acute care timeline. For example, when the ambulance picks up the patient, it may revoke access to the previous team, which is probably the call centre.

The PIP is responsible for engaging the appropriate context handlers to aggregate the relevant contextual attributes values. This is performed based on $Patient_{ID}$, which indexes the resource EMR. From the $Patient_{ID}$, the PIP can retrieve the active patient's ES_{ID}, teams involved in the ES and their timestamps, and members of each team. After acquiring the contextual attributes' values, the context handler sends them to the PDP for policy evaluation.

6.4. Policies definition phase

Following the steps in phase 4, Fig. 3, we created policies based on the rules expressed in Table 3 to protect against non-legitimate requests for accessing the patient's EMR. Table 4 summarises the policies created to read and update the patient EMR and authorise the start and end of an ES. The rules combination algorithm of each policy is defined as PERMIT unless DENY, which means that if any rule yields a DENY, the policy outcome decision will be denied. Fig. 4 represents the hierarchy of the rules on a policies decision tree.

We used AMPLE to create the rules and define the policies. Moreover, we manually defined the dynamic parameters that the context handlers use to evaluate each rule since APAM does not support this definition yet, where we create rules that both sides of the equation are parameters that the values of the contextual attributes will replace. This is obtained as follows. Consider a rule $t_{request} \geq X$, where X represents a dynamic value. The context handler will replace the parameter X for the contextual attribute value of some context value, for example, t_{invite}, which can be acquired from the PIP. Therefore, the dynamism of the access policy is introduced by design. The rule does not involve any static values since this can only be known and enforced at run-time.

Table 3
Modelling rules, request's attributes and contextual attributes involved for each entity.

Entity	Rule	Description	Logical representation
Subject	R1	The healthcare professionals are working on their shifts.	$(t_{request} \geq t_{startshift}) \wedge (t_{request} \leq t_{endshift})$
	R2	The healthcare professional must be an active member of an acute care team.	$User_{ID} \in TeamMembers$
Resource	R3	Only the EMR of the patient under ES must be available to the acute care team active in the ES.	$(Patient_{ID} \in ES\ table) \wedge (Team_{ID} \in EC\ table) \wedge (ES_{ID} \in EC\ table)$
Action	R4	The acute care team has the right to read data as soon as they are involved in the emergency session.	$t_{request} \geq t_{invite}$
	R5	The acute care team has the right to read data until they are revoked from the emergency session.	$t_{request} \leq t_{revoke}$
	R6	The acute care team has the right to add data as soon as they are in the presence of the patient.	$t_{request} \geq t_{treat}$
	R7	The acute care team has the right to add data until a predefined extra time after the treatment.	$t_{request} \leq (t_{revoke} + extratime)$
	R8	The healthcare professional from call centre or hospital acute care team has the right to start the ES.	$(Team_{tag} = Team_c) \vee (Team_{tag} = Team_h)$
	R9	The healthcare professional from the hospital acute care team has the right to end the ES, unless the healthcare professional was who started the ES.	$(Team_{tag} = Team_h) \wedge (User_{ID} \neq Starter_{ID})$

Table 4
Policy is a combination of enumerated rules according to the requested action (see Table 3)

Action	Policy
Read	R1 ∧ R2 ∧ R3 ∧ R4 ∧ R5
Update	R1 ∧ R2 ∧ R3 ∧ R6 ∧ R7
Start ES	R1 ∧ R2 ∧ R8
End ES	R1 ∧ R2 ∧ R3 ∧ R6 ∧ R9

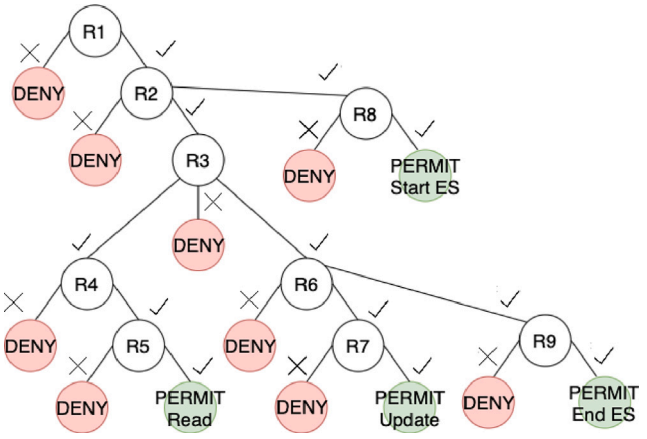

Fig. 4. Policies decision tree.

6.5. Policies enforcement phase

Following the steps in phase 5, Fig. 3, we deployed the PDP policies and added the PEP to the EMR System. After receiving the required contextual attributes and the policy evaluation, the PDP yields a decision to the context handler: PERMIT or DENY. The context handler then notifies the PEP about the decision, and – if it is a PERMIT – the PEP allows the data access request on the EMR System.

7. Implementation and evaluation

In this section, we present the implementation of a prototype and the evaluation of the policies defined in Section 6. First, we present the acute care information workflow of the AC-ABAC model. Then, through simulations, we validate the defined policies' correctness and analyse the request evaluation performance in different scenarios.

The prototype includes Contextual-Aware ABAC deployment presented in Fig. 2, a web application simulating the EMR system to be protected and a custom database that serves as PIP with a REST-API. The EMR system simulation is a web application with available endpoints that allows read, update, start ES and end ES requests. The PIP contains the attributes values that are aggregated from the EMR System. However, in this prototype, we populated the PIP database to generate attribute values from simulated interactions between users and the EMR System under emergency. The context handler uses the REST-API to retrieve and process the contextual attributes stored on the PIP. Both web application and the PIP's REST-API were developed with the Django Framework (Django, 2021).

Regarding ABAC, the open-source WSO2 Balana engine (Balana) was used as an implementation of the XACML access control. The context handlers were developed and connected to the PIP and enabled in the ABAC Enforcement Engine. With all ABAC components set up, the PEP is invoked whenever an incoming access request to a protected resource is detected, and the evaluation process begins. The Docker image of the prototype, the defined policies and context handlers of the AC-ABAC model, and the results of the experiments can be found on GitHub (de Oliveira et al., 2021a).

7.1. Acute care information workflow

Here we describe the acute care information workflow used in the AC-ABAC model. Guided by the methodology, we understood when and which information we infer during the acute care workflow. Fig. 5 presents an ES flow where the call centre team starts the ES and invites an ambulance team, the ambulance team invites a hospital team, and the hospital team ends the ES. Each team has a starting point that represents the moment when the team enters the ES. The acute care teams interact with the ABAC engine to obtain access rights, and a team member notifies the EMR system about the events on the patient's ES. The access permissions are granted for the teams during a period of time and are updated according to the subsequent events on the patient's ES. Leave ES represents when an EC of the team ends, while an End ES represents the end of the entire ES. The use case can be extended to support more teams participating in the ES, for example, when the patient needs an ambulance transfer to a second hospital.

7.2. Correctness evaluation

In this section, we present the simulation results to demonstrate the correctness of the policies defined in Section 6, Fig. 4. This is done by evaluating the policy implementation with a test input (i.e., access

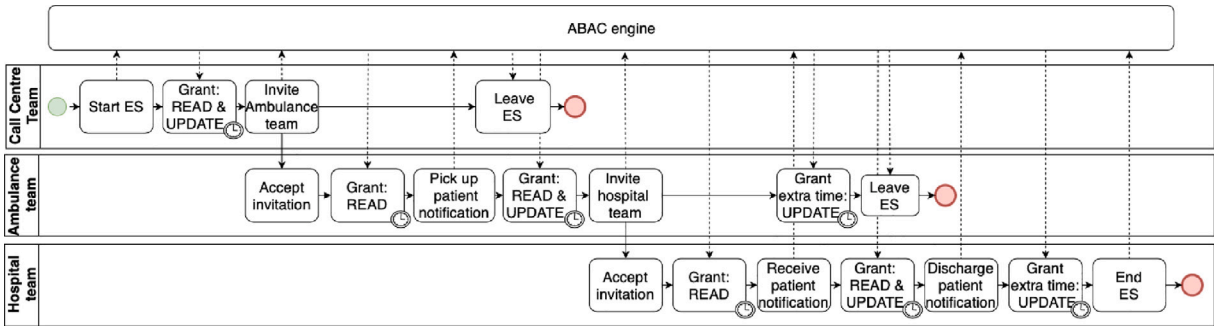

Fig. 5. Emergency session flow presented as Business Process Model and Notation (BPMN) diagram. The teams have access rights granted to read and update according to the timeline of events.

Table 5
Description of each scenario tested in the experiments.

Scenarios	Description
S1	A team member in the ES requests to read the EMR when $t_{invite} \leq t_{request} \leq t_{treat}$
S2	A team member in the ES requests to read the EMR when $t_{treat} \leq t_{request} \leq t_{revoke}$
S3	A team member in the ES requests to update the EMR when $t_{treat} \leq t_{request} \leq t_{revoke}$
S4	A team member in the ES requests to update the EMR when $t_{revoke} \leq t_{request} \leq (t_{revoke} + extratime)$
S5	A professional is not active on the shift, but is a team member participating in the ES
S6	A professional is active on the shift but is not currently a team member
S7	A professional is active on the shift and is a team member, but the team is not part of the ES
S8	A professional is active on the shift is a team member and is part of the ES, but requests another patient's EMR.
S9	A team member in the ES requests to read the EMR when $t_{request} > t_{revoke}$
S10	A team member in the ES requests to update the EMR when $t_{request} < t_{treat}$
S11	A team member from the ES requests to update the EMR when $t_{request} > (t_{revoke} + extratime)$
S12	A team member from a call centre or hospital team requests to start an ES
S13	A team member from an ambulance team requests to start an ES
S14	A team member from the hospital team requests to end an ES and $User_{ID} \neq Starter_{ID}$
S15	A team member from the hospital team requests to end an ES and $User_{ID} = Starter_{ID}$

Table 6
Policies correctness evaluation in different scenarios. The outcome represents the obtained decision after the policy evaluation.

Scenario	Rules									Action	Outcome
	R1	R2	R3	R4	R5	R6	R7	R8	R9		
S1	✓	✓	✓	✓	✓	–	–	–	–	Read	PERMIT
S2	✓	✓	✓	✓	✓	–	–	–	–	Read	PERMIT
S3	✓	✓	✓	–	–	✓	✓	–	–	Update	PERMIT
S4	✓	✓	✓	–	–	✓	✓	–	–	Update	PERMIT
S5	X	✓	✓	–	–	–	–	–	–	Read & Update	DENY
S6	✓	X	✓	–	–	–	–	–	–	Read & Update	DENY
S7	✓	✓	X	–	–	–	–	–	–	Read & Update	DENY
S8	✓	✓	X	–	–	–	–	–	–	Read & Update	DENY
S9	✓	✓	✓	✓	X	–	–	–	–	Read	DENY
S10	✓	✓	✓	–	–	X	✓	–	–	Update	DENY
S11	✓	✓	✓	–	–	✓	X	–	–	Update	DENY
S12	✓	✓	–	–	–	–	–	✓	–	Start ES	PERMIT
S13	✓	✓	–	–	–	–	–	X	–	Start ES	DENY
S14	✓	✓	✓	–	–	✓	–	–	✓	End ES	PERMIT
S15	✓	✓	✓	–	–	✓	–	–	X	End ES	DENY

request) and validating the corresponding output (i.e., PERMIT or DENY).We have simulated legitimate and non-legitimate requests in different scenarios in the prototype to evaluate the policies' correctness. Table 5 presents the description of fifteen scenarios that were evaluated (S1-15), and their expected and obtained outcomes are summarised in Table 6. Some scenarios may be unrealistic because we evaluated each policy's different rules and outcomes to demonstrate that the security mechanism works.

Scenarios 1–4 consist of must-be-permitted access requests to the patient's EMR. Scenarios 5–11 are must-be-denied access requests to the patient's EMR. Finally, scenarios 12–15 describe the requests to start and end an ES. Table 6 presents the policies per action as the combination of rules evaluated in each scenario. It also indicates the rules that fail in each of the must-be-denied scenarios. Table 6 also presents the outcomes obtained with the simulation, which corresponded to the expected values in all cases.

We acknowledge that false-positive and false-negative results might happen due to race conditions. For example, while the PIP updates the contextual attributes, the evaluation might consider outdated contextual attributes. Note, however, that this hardly occurs since the database updates in a matter of milliseconds.

7.3. Performance evaluation

Using our application simulating the EMR system, we implemented a Python script that simulates requests issued by a "Requester" through the application client to the EMR system server. The Context-Aware ABAC prototype intercepts all the requests to the EMR system server and yields a response. Here we assess the performance of the AC-ABAC model implementation regarding the time needed to evaluate an access request, evaluate the policy, and deliver the response (permit or deny).

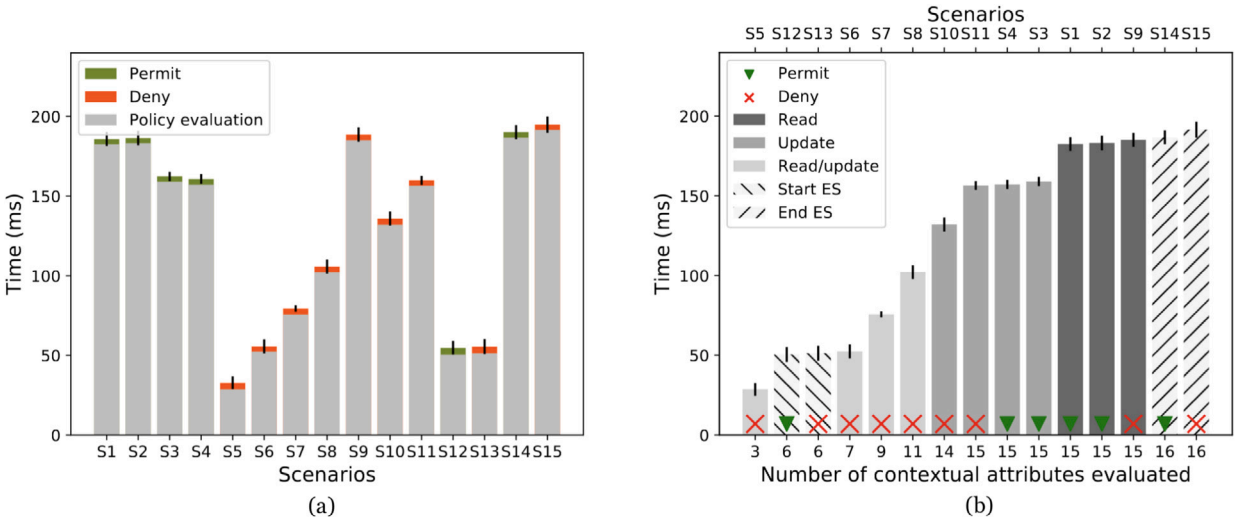

Fig. 6. Time to process access requests in each scenario (average time for 100 repetitions; error bars represent the confidence interval of 95% around the averages. a) Request and policy evaluation time. Must-be-permit scenarios in green and must-be-deny in red, policy evaluation times in grey. b) Policy evaluation time per number of attributes evaluated. The policies are differentiated per action.

The requests implemented the fifteen scenarios presented in Section 7.2. Table 5 and repeated 100 times. We measured the total time since the PEP received the request until it delivered the response (permit or deny) and the time dedicated inside the PDP for the policies' evaluation. The experiments were executed in a DELL PowerEdge R630 server with an Intel Xeon ES-2640 v4 Processor and 256 GB of RAM at the cloud of the University of Westminster (DesLauriers et al., 2021).

Fig. 6 presents the results of the performance experiment. Fig. 6(a) presents the total time to process the request in each scenario, as well as the portion dedicated to policy evaluation (average and standard deviation calculated over 100 runs). As expected, in grey, the time for evaluating the policy dominates the total request evaluation time. This happens because the time spent inside the PDP includes the waiting time for retrieving the policy from the PAP and the contextual attributes from the context handlers. Each scenario presents a different request evaluation time because they use different policies. Moreover, the policies are defined as "permit unless denied", and the rules are evaluated in sequence, as listed in Fig. 4. So, when one of the rules results in a denial, it stops the execution of the subsequent rules. Note that the scenarios that evaluated the same rules presented similar average request evaluation times (see details in Table 6). The exceptions are scenarios S5–S8 and S10, which did not evaluate all the policy rules. In each of these scenarios, a different rule causes the request to be denied, so a different number of rules, and the necessary attributes, are evaluated — this results in different processing times. Fig. 6(b) shows how the number of evaluated contextual attributes can affect policy evaluation time. Note that the policy evaluation time increases when more attributes need to be evaluated, as expected. However, the result also shows that the number of attributes is not the only factor affecting policy evaluation time. For example, although in S1 and S3, the number of evaluated attributes is the same (fifteen), policy evaluation takes longer in S1. This indicates that different contextual attributes might require different efforts to be inferred.

Regarding the results, the longest average time to evaluate a request was 194.89 ms for S14 and S15. In both scenarios, the user requests to end an ES, which is the most complex policy of our modelling because its evaluation involves the validation of five rules that combine sixteen contextual attributes. However, start and end ES requests happen only once per patient, while reading and updating requests happen more frequently. The average time to evaluate a request to read was 185.82 ms in S1, and 162.36 ms for a request to update in S3. We suggest that these times to process such complex requests are acceptable, particularly considering the security improvement added to the system when using more fine-grained and context-aware access control policies. This performance is also acceptable with other security-related delays that may include data encryption and decryption (Bakas, Dang, Michalas, & Zalitko, 2020). Finally, the obtained performance is also similar to other ABAC approaches (Verginadis et al., 2019). For example, a simple WSO2 Balana-based execution of ABAC policies can reach 187 ms on average for yielding an access control decision. Although more sophisticated approaches rely on XACML and semantic inferencing for yielding access control decisions, they can even exceed 8000 ms, as the policy evaluation time exponentially increases depending on the number of contextual attributes involved in the deployed ABAC rules (Verginadis et al., 2019).

8. Discussion

Through the proposed methodology, we followed the steps needed to understand the access control dynamism required for this acute care application. The preparation phase of the methodology facilitated collecting the requirements and understanding the stakeholders in a structured manner imposed by the templates. In the second analysis phase, we had to make a choice of contextual attributes to be used to create the access control policies. To guarantee the availability of a patient's EMR at all times, we decided to leave out contextual attributes regarding location, such as GPS coordinates and IP addresses. The iterative approach adopted in the development phase helped refine realistic rules and contextual attributes. In the policies definition phase, we noticed that the AMPLE editor does not define rules with both parameters as contextual attributes. Therefore, to create the rules, we manually modified the XACML rules after creating them on the AMPLE editor. We plan to explore the possibility of direct creating dynamic parameters for those rules on the AMPLE editor in future work. Finally, during the policies enforcement phase, we observed that race conditions regarding outdated security tokens might occur. Such inconsistencies can be minimised by regenerating tokens frequently after modifying the team composition.

Regarding patient identification, AC-ABAC model assumes that the patient is registered beforehand, so there is a $\text{Patient}_{\text{ID}}$. However, it

is possible that the patient cannot be found or identified during the emergency (e.g. unconscious patient). In such cases, the EMR system should give a temporary identification (e.g. 'John Doe' or 'Joanna Doe') to the patient so that during the ES, the teams can share the collected information. The proper registration or attribution of the episodes of care can be done after the ES has ended.

Furthermore, we highlight that the defined policies can dynamically change at run-time without any need to re-compile or restart the authorisation engine. For example, imagine that during the COVID-19 pandemic, there was a shortage of ambulances due to many people going to the hospital. In such a case, the paramedic teams of the military forces could provide emergency response. The proposed model could be instantly updated with a policy to add the military teams without compromising the rest of the operational policies in the system.

Regarding the experiment results, the different times found for the request evaluation in the various scenarios indicate that the access control system could be susceptible to timing attacks (Brumley & Tuveri, 2011). In a timing attack, the attacker tries to discover vulnerabilities in the security of a system by studying the variation in its response time to different input parameters. In the case of ABAC, an attacker could analyse the response time according to his attributes and discover which are correct because they lead to a longer response time. Moreover, the attacker could identify the policy by knowing which correct attributes and perform exploitation on the access control. A solution for this is simply to answer all the requests with nearly-constant time.

9. Conclusions

This paper presented an access control modelling that keeps patient data confidential without compromising the data availability for the legit acute care teams. Firstly, we introduced a step-by-step methodology that leveraged fine-grained access control modelling using the Context-Aware Attribute-Based Access control (ABAC) model in an EMR system. We understood which rules and contextual attributes should and should not be used to legitimate access to the patient's EMR for the acute care teams during an emergency session through an analysis guided by the proposed methodology. Secondly, we developed the Acute Care Attribute-Based Access Control (AC-ABAC) model, which has access policies and contextual attributes to enable granting and revoking access to legit healthcare professionals. Finally, we developed a prototype of the EMR system integrated with the Contextual-Aware ABAC engine to explore multiple scenarios and evaluate the correctness and performance of the AC-ABAC model.

Furthermore, we implemented a prototype of AC-ABAC model in a use case. Finally, we evaluated the prototype in multiple scenarios to check the policies' correctness. In all scenarios, the outcome resulted as expected. Using the prototype, we simulate requests for various scenarios and evaluate the performance of our AC-ABAC modelling. The results show that the longest average time to process a request was 194.89 ms, which we consider reasonable when comparing ABAC with other security-related delays, including data encryption and decryption. Moreover, we suggest that the time added to the overall request process is worthwhile, considering the security added to the system. The proposed AC-ABAC model enables the patient's EMR availability for the acute care teams considering a real-case emergency scenario and the complexity of multiple team collaboration without neglecting the patient's EMR security and privacy requirements.

CRediT authorship contribution statement

Marcela T. de Oliveira: Conceptualization, Methodology, Formal analysis, Investigation, Writing — original draft, Writing – review & editing. **Yiannis Verginadis:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Lúcio H.A. Reis:** Software, Investigation, Writing – review & editing, Formal analysis, Validation. **Evgenia Psarra:** Software, Investigation. **Ioannis Patiniotakis:** Software. **Sílvia D. Olabarriaga:** Supervision, Conceptualization, Resources, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my code as reference in the manuscript.

Acknowledgements

This work was funded by the ASCLEPIOS project (Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare) of the European Union's Horizon 2020 research and innovation program under grant agreement No. 826093.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2022.119271.

References

Abbas, A., & Khan, S. U. (2014). A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. *IEEE Journal of Biomedical and Health Informatics*, *18*(4), 1431–1441. http://dx.doi.org/10.1109/jbhi.2014.2300846.

Abomhara, M., Yang, H., & Køien, G. M. (2016). Access control model for cooperative healthcare environments: Modeling and verification. In *2016 IEEE international conference on healthcare informatics* (pp. 46–54). IEEE, http://dx.doi.org/10.1109/ichi.2016.10.

Arora, A., & Gosain, A. (2020). Dynamic trust emergency role-based access control (DTE–RBAC). *International Journal of Computer Applications*, *175*(24), 0975–8887. http://dx.doi.org/10.5120/ijca2020920773.

Bakas, A., Dang, H. -V., Michalas, A., & Zalitko, A. (2020). The cloud we share: Access control on symmetrically encrypted data in untrusted clouds. *IEEE Access*, *8*, 210462–210477. http://dx.doi.org/10.1109/access.2020.3038838.

Balana (2014). WSO2 balana. https://github.com/wso2/balana. (Accessed 11 March 2021).

Bhuyan, S. S., Kabir, U. Y., Escareno, J. M., Ector, K., Palakodeti, S., Wyant, D., et al. (2020). Transforming healthcare cybersecurity from reactive to proactive: Current status and future recommendations. *Journal of Medical Systems*, *44*(5), 1–9. http://dx.doi.org/10.1007/s10916-019-1507-y.

Brumley, B. B., & Tuveri, N. (2011). Remote timing attacks are still practical. In V. Atluri, & C. Diaz (Eds.), *Computer security – ESORICS 2011* (pp. 355–371). Berlin, Heidelberg: Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-23822-2_20.

Calvillo-Arbizu, J., Román-Martínez, I., & Roa-Romero, L. M. (2014). Standardized access control mechanisms for protecting ISO 13606-based electronic health record systems. In *IEEE–EMBS international conference on biomedical and health informatics*. IEEE, http://dx.doi.org/10.1109/bhi.2014.6864421.

DesLauriers, J., Kiss, T., Ariyattu, R. C., Dang, H. -V., Ullah, A., Bowden, J., et al. (2021). Cloud apps to-go: Cloud portability with TOSCA and MiCADO. *Concurrency Computations: Practice and Experience*, *33*(19), http://dx.doi.org/10.1002/cpe.6093.

Django (2021). Django framework. https://www.djangoproject.com/. (Accessed 11 March 2021).

European Commission (2016). What is a data controller or a data processor? https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/obligations/controller-processor/what-data-controller-or-data-processor. (Accessed 8 December 2020).

European Union (2016). European parliament and council of European union regulation (EU) 2016/679. https://gdpr-info.eu. (Accessed 16 December 2020).

Ferreira, A., Chadwick, D., Farinha, P., Correia, R., Zao, G., Chilro, R., et al. (2009). How to securely break into RBAC: The BTG–RBAC model. In *2009 annual computer security applications conference*. http://dx.doi.org/10.1109/ACSAC.2009.12.

FHIR (2011). FHIR episode of care. https://www.hl7.org/fhir/episodeofcare.html. (Accessed 11 March 2021).

Hillestad, R., Bigelow, J., Bower, A., Girosi, F., Meili, R., Scoville, R., et al. (2005). Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health Affairs (Project Hope)*, *24*(5), 1103–1117. http://dx.doi.org/10.1377/hlthaff.24.5.1103.

Lunardelli, A., Matteucci, I., Mori, P., & Petrocchi, M. (2013). A prototype for solving conflicts in XACML-based e-health policies. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems*. http://dx.doi.org/10.1109/cbms.2013.6627838.

Michalas, A. (2019). The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 146–155). http://dx.doi.org/10.1145/3297280.3297297.

Mitra, B., Sural, S., Vaidya, J., & Atluri, V. (2016). A survey of role mining. *ACM Computing Surveys*, *48*(4), 1–37. http://dx.doi.org/10.1145/2871148.

Nazerian, F., Motameni, H., & Nematzadeh, H. (2019). Emergency role-based access control (E-RBAC) and analysis of model specifications with alloy. *Journal of Information Security and Applications*, *45*, 131–142. http://dx.doi.org/10.1016/j.jisa.2019.01.008.

OASIS (2020a). Extensible access control markup language (XACML) version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. (Accessed 23 November 2022).

OASIS (2020b). OASIS XACML technical committee. https://www.oasis-open.org. (Accessed 23 November 2022).

de Oliveira, M. T., Bakas, A., Frimpong, E., Groot, A. E., Marquering, H. A., Michalas, A., et al. (2020). A break-glass protocol based on ciphertext-policy attribute-based encryption to access medical records in the cloud. *Annals of Telecommunications*, *75*(3), 103–119. http://dx.doi.org/10.1007/s12243-020-00759-2.

de Oliveira, M. T., Verginadis, Y., Reis, L. H. A., Evgenia Psarra, I. P., & Olabarriaga, S. D. (2021a). AC–ABAC github repository. https://github.com/AMCeScience/AC-ABAC-modelling-public. (Accessed 1 April 2021).

de Oliveira, M. T., Verginadis, Y., Reis, L. H. A., Evgenia Psarra, I. P., & Olabarriaga, S. D. (2021b). AC–ABAC templates. https://github.com/AMCeScience/AC-ABAC-modelling-public/tree/main/templates. (Accessed 1 April 2021).

Peleg, M., Beimel, D., Dori, D., & Denekamp, Y. (2008). Situation-based access control: Privacy management via modeling of patient data access scenarios. *Journal of Biomedical Informatics*, *41*(6), 1028–1040. http://dx.doi.org/10.1016/j.jbi.2008.03.014.

Psarra, E., Verginadis, Y., Patiniotakis, I., Apostolou, D., & Mentzas, G. (2020a). A context-aware security model for a combination of attribute-based access control and attribute-based encryption in the healthcare domain. In *Workshops of the international conference on advanced information networking and applications* (pp. 1133–1142). Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-44038-1_104.

Psarra, E., Verginadis, Y., Patiniotakis, I., Apostolou, D., & Mentzas, G. (2020b). Securing access to healthcare data with context-aware policies. In *11th international conference on information, intelligence, systems and applications* (pp. 1–6). http://dx.doi.org/10.1109/iisa50023.2020.9284393.

Rao, K. R., Nayak, A., Ray, I. G., Rahulamathavan, Y., & Rajarajan, M. (2021). Role recommender-RBAC: Optimizing user-role assignments in RBAC. *Computer Communications*, *166*, 140–153. http://dx.doi.org/10.1016/j.comcom.2020.12.006.

Seol, K., Kim, Y. -G., Lee, E., Seo, Y. -D., & Baik, D. -K. (2018). Privacy-preserving attribute-based access control model for XML-based electronic health record system. *IEEE Access*, *6*, 9114–9128. http://dx.doi.org/10.1109/access.2018.2800288.

Verginadis, Y., Michalas, A., Gouvas, P., Schiefer, G., Hübsch, G., & Paraskakis, I. (2017). PaaSword: A holistic data privacy and security by design framework for cloud services. *Journal of Grid Computing*, *15*(2), 219–234. http://dx.doi.org/10.1007/s10723-017-9394-2.

Verginadis, Y., Patiniotakis, I., Gouvas, P., Mantzouratos, S., Veloudis, S., Schork, S. T., et al. (2019). Context-aware policy enforcement for PaaS-enabled access control. *IEEE Transactions on Cloud Computing*, *10*(1), 276–291. http://dx.doi.org/10.1109/tcc.2019.2927341.

XSD (2012). W3C XML schema definition language (XSD). https://www.w3.org/TR/xmlschema11-1/. (Accessed 8 December 2020).