GRADUATION: A GDPR-based Mutation Methodology

Said Daoudagh[1,2] [0000−0002−3073−6217] and Eda Marchetti[1] [0000−0003−4223−8036]

[1] ISTI-CNR, Pisa, Italy
{said.daoudagh, eda.marchetti}@isti.cnr.it
[2] University of Pisa, Pisa, Italy

Abstract. The adoption of the General Data Protection Regulation (GDPR) is enhancing different business and research opportunities that evidence the necessity of appropriate solutions supporting specification, processing, testing, and assessing the overall (personal) data management. This paper proposes GRADUATION (GdpR-bAseD mUtATION) methodology, for mutation analysis of data protection policies test cases. The new methodology provides generic mutation operators in reference to the currently applicable EU Data Protection Regulation. The preliminary implementation of the steps involved in the GDPR-based mutants derivation is also described.

Keywords: Data Protection · GDPR · Mutation Operators · Privacy Policies · Security Policies.

1 Introduction

The widespread adoption of the General Data Protection Regulation (GDPR), i.e., the EU Data Protection Regulation [12], if on the one hand is enhancing different business and research opportunities within the Information and Communication Technology (ICT) environment, on the other hand is struggling in the definition of appropriate procedures and technical solutions for specifying the privacy requirements, processing personal data, and testing the overall data management. Indeed, privacy legislation's requires to deploy adequate fine-grained mechanisms that are able to continuously enforce and verify legal requirements, such as the data usage purpose, the user consent and the data retention period. To this purpose, different proposals are currently available for automatic defining, implementing and testing privacy knowledge and rules [1, 3, 7, 23, 24], but few attention is still devoted to the assessment of the testing suites or strategies adopted for validating the different GDPR implementation aspects. Indeed, the fault detection effectiveness is a fundamental parameter for ensuring the quality properties of the final products and for prioritizing and/or selecting test cases for regression testing activities [15]. To this purpose, one of the most adopted approaches is the Mutation testing, i.e., a technique in which syntactic faults, simulating typical programmer's mistakes, are seeded in the original program in order to produce a set of faulty programs, called mutants, each containing one

fault. Therefore, a predefined set of test cases is executed both on the original program and its mutants, and outputs collected: if the mutant's output is different from the original program's one, the fault is detected and the mutant is said to be killed. The mutation score is the ratio of the number of detected faults over the total number of seeded faults and indicates the effectiveness of the test suite.

In the context of the GDPR, and data privacy management in general, only few proposals are targeting the definition of mutation operators able to deal with the specific privacy characteristics and requirements of the privacy standards [2]. In these cases, the proposed mutation operators do not exhaustively cover all the important criticalities of the GDPR. For instance, they do not consider mutation operators concerning the erroneous use of the purpose defined by the controller and the consent given by the data subject.

In this paper, we move a step ahead in this research direction by presenting the new GdpR-bAseD mUtATION (GRADUATION) methodology, partially supported by a prototype tool, for: 1. analysing and managing model-based specifications of legal text (such as the GDPR), so as to extract main concepts and useful data; 2. selecting and applying a set of mutation operators to a specific GDPR-based model instance, so as to derive its mutated versions.

To better clarify the methodology application, we present the specialization of the GRADUATION in the context of GDPR-based authorization systems. Indeed, privacy legislation requires organizations to deploy adequate fine-grained Access Control (AC) mechanisms [14] that take into account additional legal requirements, such as the data usage purpose, the user consent and the data retention period. Consequently, this rises up the problem of developing effective and efficient test strategies able to guarantee the lack of unauthorized access to personal data (*security perspective*) and unlawful processing (*legal perspective*).

It is important to notice that even if the specialization of the GRADUATION tool refers to the AC mechanisms based on the Attribute-Based Access Control (ABAC) model [14], the GRADUATION methodology, and in particular its mutation operators set, is agnostic with respect to the AC mechanisms specification language, and can be applicable to any system that dealing with the GDPR.

In this paper, with the aim of providing a comprehensive assessment environment, the specialization of the GRADUATION presented includes: 1) all the currently available AC-based mutation operators [8], i.e., the traditional ABAC mutation operators [8]; and 2) the new conceived operators based on the GDPR's peculiarities.

Summarizing, the main contributions of this paper are:

- a generic methodology, called GRADUATION for automatically generating GDPR-based mutants;
- a set of GDPR-based mutation operators focusing on the GDPR's peculiarities;
- a preliminary implementation of the steps involved in application of mutation operators and mutants generation in the ABAC context; and

– an example of application GRADUATION methodology in the ABAC context.

The objective therefore is to define an abstract process for the automatic generation of GDPR-based mutants, useful for assessing generic GDPR-based testing strategies through mutation analysis. Indeed, the derived test suites can be used by:

1. The Controller to assess the GDPR's readiness of the Processor, i.e., the responsible of the Personal Data processing.
2. The Supervisor Authority for verifying the GDPR compliance of the processes defined by the Controller; and
3. The Data Protection Officer (DPO) for ensuring that the organisation processes Personal Data in compliance with the data protection rules.

Outline. Section 2 presents the background knowledge about the GDPR, ABAC and mutation analysis; whereas, Section 3 illustrates related work. We present GRADUATION methodology in Section 4, and the specific GDPR-based mutation operators in Section 5. In Section 6, the implementation of the steps involved in mutants generation are presented, while an example of the application of GRADUATION methodology is provided in Section 7. Finally, Section 8 concludes the paper by also hinting at future work.

2 Background

In this section, we briefly provide an overview of the GDPR and ABAC model. Other basic concepts, useful for understanding the proposal are provided in line within the text in Section 4.

GDPR Concepts. The GDPR [12] defines Personal Data as any information relating to an identified or identifiable natural person called Data Subject. In this view the, a Data Subject is a Natural Person and her/his data are managed by a *Controller*. The GDPR rules the processing of personal data, whether it is automated (even partially) or not. The GDPR relies on the following principles and demands: *Purposes*, i.e., data should only be collected for determined, explicit and legitimate purposes, and should not be processed later for other purposes; *Accuracy*, i.e., the processed data must be accurate and up-to-date regularly; *Retention*, i.e., data must be deleted after a limited period; *Subject explicit consent*, i.e., data may be collected and processed only if the data subject has given her or his explicit consent. The most adopted model-based representation of the GDPR relies on ontologies [4, 25, 16, 20, 21].

ABAC and Mutation. ABAC [14] is currently one of the mostly adopted AC model in industrial environment [13] and "supplements and subsumes" the other models [14]. The National Institute of Standards and Technology (NIST) defines ABAC as "[a]n access control method where subject requests to perform

operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions" [13]. In the recent years, several proposals used ABAC model to represent GDPR's concepts [5, 6, 10, 11], by casting the conceived representation into the eXtensible Access Control Markup Language (XACML) standard [19]. Indeed, XACML is the only available standard implementation of ABAC model, and it is a platform-independent XML-based language for the specification of Access Control Policies (ACPs). The main purpose of an XACML policy is to define the constraints that a subject (e.g., Data Subject or Controller) needs to comply with for accessing a resource (e.g., Personal Data) and doing an action (e.g., a processing activity) in a given environment (e.g., purpose and consent). However, as stated in the previous section, developing XACML-based ACPs rises up the problem of developing effective and efficient test strategies able to guarantee the lack of unauthorized access to personal data (*security perspective*) and unlawful processing (*legal perspective*). Therefore, mutation analysis [22] can be applied on ACPs for measuring the adequacy of the generated test suites. The general process of mutation analysis consists of two steps: first, change the original program (e.g., ACP) with predefined mutation operators and generates a set of mutated program, called mutants; then, the mutants are executed against a test suite, and information is collected during the execution for various purpose of analysis.

3 Related Work

In the context of the GDPR, and data privacy management in general, only few proposals are targeting the definition of mutation operators able to deal with the specific privacy characteristics and requirements of the privacy standards [2]. And in these cases, the proposed mutation operators do not exhaustively cover all the important criticalities of the GDPR. For instance, they do not consider mutation operators concerning the erroneous use of the purpose defined by the controller and the consent given by the data subject.

Focusing in particular on mutation testing in the context of access control, the most noteworthy proposals are: the fault models and relative set of mutation operators simulating syntactic faults of XACML access control policies proposed by [17]; the generic metamodel for the specification rule-based security policy and the relative set of mutation operators provided by [18]; the XACMUT tool [8], which includes and enhances the mutation operators of [17] and [18] addressing specific faults of the XACML 2.0 language; and the proposal of [9] which implements mutation analysis at the level of the policy evaluation engine instead of applying it at the level of access control policy.

On the contrary, considering the mutation testing in the context of the GDPR, to the best of our knowledge the only proposal currently available is represented by [2]. Indeed, this paper is the first attempts of extending mutation operators for validating ontologies expressing the GDPR's provisions. However,

even if generic, the mutation operators proposed in the paper do not cover all the specific aspects of the privacy standard.

Therefore, our proposal on the one hand extends the set of mutation operators, so as to validating the test suites or strategies against the GDPR peculiarities, on the other provides an implementation able to integrating into a unique environment all the existing approaches for mutation testing in the area of access control system.

4 Methodology for GDPR-based Mutants Derivation

GRADUATION methodology is composed of four main steps (see Figure 1): (1) Model Derivation; (2) Model Parsing; (3) Implementation Parsing; and (4) Mutation Application.

Although grounded in a domain-related implementation (i.e., the GDPR), GRADUATION yields a more general spectrum, since it can be applied to different data protection regulations and more in general to any legal text that implicitly contains, or suggests, data protection requirements. In the following, details about the methodology steps are provided.

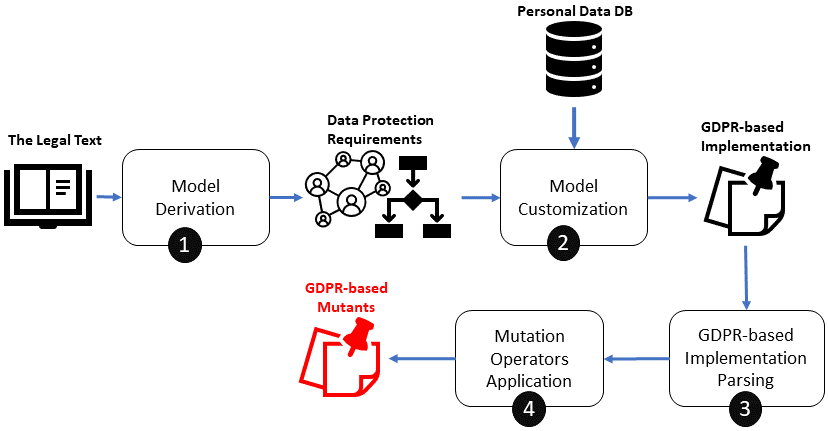

Fig. 1. GDPR-based Mutation Methodology.

Model Derivation (Step (1)). Starting from a legal text, in our case the GDPR, the model representing the main concepts and the relations between them is obtained. To this purpose, in literature different proposals focused on the derivation of a formal representation of legal text are available [4, 25, 16, 20, 21]. It is out of the scope of this work investigating the most suitable approaches for this purpose. The hypothesis of our work is that a GDPR-based model is available in terms of a specification language, for instance an ontological representation, a UML model or an access control model.

Model Customization(Step ②). This step takes as input both the legal text model and concrete inputs stored in *Personal Data DB*. In this step the legal-based model (in our case the GDPR-based model) is analyzed to identify main legal concepts and associate to each of them the proper input domain (i.e., the data contained in *Personal Data DB*). Examples of legal data could be: personal data, data subject, controller, processor, consent and purpose. The legal-based model and the identified input are then used for deriving a specialized GDPR-based implementation of the considered model.

GDPR-based Implementation Parsing (Step ③). According to the mutation testing approach, the derived GDPR-based implementation is classified as *gold* implementation, and it is used for: i) identifying the set of data entities, such as for instance the current ID of the Processor, the name of a Data Subject and so on; ii) instrumenting the *gold* implementation so as to let the automatic derivation of its mutated versions (i.e., mutants set).

Mutation Operators Application (Step ④). The set of GDPR-based mutation operators is applied to the gold implementation so as to derive the mutants set. In this step, two kinds of mutations are considered: *intra-implementation* and *inter-implementation mutations*. The former set refers to the application of mutation operators managing only the information and data extracted from the *gold* implementation (i.e., during Step ③). The latter set refers to mutation operator managing the information relative to the GDPR-based model (i.e., the model derived during the Step ②).

The conceived GDPR-based mutation operators are reported in Section 5.

5 GDPR-based Mutation Operators

The GDPR-based mutation operators can be classified in three main categories:

1. operators targeting the purpose of processing and the consent given by data subject;
2. mutation operators targeting the roles defined in the GDPR such as Data Subject, Controller and Processor; and finally,
3. operators focusing on Personal Data, i.e., the object of the EU legal framework, and their categories.

These operators have the ability to be applied to different domains, because voluntarily conceived as generic. Therefore, depending on the specific language or formalism used for defining the GDPR's requirements, they can be implemented and applied accordingly.

The new generic GDPR-based mutation operators are as in the following:

Giving Consent (GC) this operator changes the value of the Consent given by the data subject.

Withdraw Consent (WC) this operator is dual to GC, and it changes the value of the consent element in the targeted implementation.

Change Purpose (CP) this operator replaces a purpose with other defined in the considered implementation. In case there is only one purpose, CP operator changes the purpose with a random one defined in the GDPR model or in other available supporting sources.

Change Controller (CC) this operator replaces a Controller with another one. In case missing candidates, CC changes the current Controller with a randomly generated Controller. This operator is applied also when Joint Controllers exit and involved in the processing of Personal Data, i.e., in defining the Purpose of processing, obtaining the consent and using Personal Data accordingly.

Replace Data Subject (RDS) this operator is able to replace a Data Subject with another one. Similar to CC and CP operators, i.e., in case of missing candidates, RDS chooses random Data Subject that replaces the current one.

Replace Controller with Processor (RCP) this operator changes a Controller with a Processor presented in the current implementation;

Replace Processor (RP) this operator replaces a Processor with another Processor.

Change Personal Data (CPD) this operator is able to change a personal data with another one.

Change Personal Data Category (CPDC) this operator changes the category of given personal data with another one.

6 GDPR-based Mutation Operators Implementation

In this section, we describe the contextualization of GRADUATION methodology in the context of ABAC-based systems. For this purpose, steps of the GRADUATION methodology (see Section 4) have been divided into three modules: Module Ⓐ refers to the activities for modeling the GDPR and deriving the ABAC policies (Steps ① and ② of Figure 1); Module Ⓑ refers to the ABAC policy parsing (Step ③ of Figure 1); and Module Ⓒ contains specific activities for the mutation testing application (Step ③ of Figure 1).

In this section, however, we only provide the implementation of Modules Ⓑ and Ⓒ, which are the most specific for the ABAC context[3].

As reported in Figure 2, Module Ⓑ is implemented as parser of the ABAC policy. Its role is to extract the data for deriving the mutated versions of the ABAC policy. An example of the information collected by this parser is provided in Table 1 of the following section.

Module Ⓒ is composed of the following components:

- *Mutation Operators Selector (Component @)*: this component implements two set of mutation operators: (i) the GDPR-based Mutation Operators defined in the previous section; and (ii) Standard ABAC Mutation Operators.

[3] GRADUATION has been implemented in Java, and it is currently available at: http://security.isti.cnr.it/tools/graduation

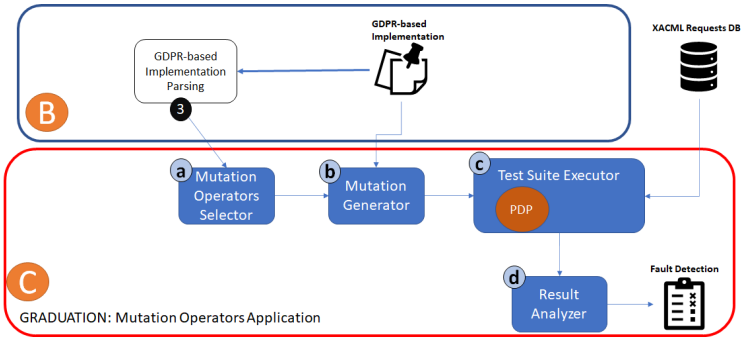

Fig. 2. Overview of GRADUATION.

This last set of operators can be categorized based on the ABAC policy elements. There are operators emulating fault at: (1) Poliy Set element level such as Policy Set Target True (PSTT), Policy Set Target False (PSTF) and Change Policy Combining Algorithm (CPC); (2) Policy element level, e.g., Change Rule Combining Algorithm (CRC) and Policy Target False (PTF); (3) Rule element level, such as Rule Target True (RTT), Rule Condition False (RCF) and Change Rule Effect (CRE); and finally, (4) Policy Functions level, for instance RemoveUniquenessFunction (RUF), ChangeLogicalFunction (CLF) and AddNotFunction (ANF). For a more detailed description and comprehensive overview of the standard AC mutation operators, we refer the reader to [8].

- *Mutants Generator (Component ⓑ)*: this component has the responsibility of generating mutated versions of the Gold (GDPR-based) policy by applying the selected mutation operators (both standard and GDPR-based) by end-user.
- *Test Suite Executor (Component ⓒ)*: this component executes the AC requests provided by the user on the original GDPR-based ABAC policy (Gold Policy) and on the generated set of mutated policies. For requests evaluation this component integrates an ABAC PDP engine, which is able to provide the corresponding result (Permit, Deny, NotApplicable or Indeterminate) for a given policy P and a request Req.
- *Results Analyzer (Component ⓓ)*: this component takes as input the results obtained by the execution of the test suites on the original GDPR-based policy and on its set of mutants, and computes the fault detection effectiveness. It works as follows: for each request the result obtained by its execution on the original policy is compared with that obtained on its mutants set. If the results are different, the mutant is classified as killed. The component provides as output the list of killed mutants, survived mutants, and the percentage of fault detection effectiveness obtained by the requests execution. It also provides functionalities allowing to filter by mutation operators, by test cases, and by the expected authorization decision. This is useful for provid-

ing different perspective of the data and for analyzing deeply the different aspects of these mutation data views.

For the aim of completeness, even if out of the scope of this paper, components ⓒ and ⓓ have been described in this section because part of the implementation of Module Ⓒ. However, they will be not further detailed in the reminder of this paper.

7 Using GRADUATION Methodology

In this section, we briefly detail the application GRADUATION methodology (see Section 4) by considering a use case scenario concerning a fitness environment taken from the literature [10]. Specifically, we consider the situation in which Alice, a Data Subject, wants to use a smart fitness application to monitor her daily activities to achieve a predefined training objective. In this case, we suppose that a customized (mobile) application is provided by a generic myFitness company (the Controller). To meet Alice's needs, myFitness has so far defined two purposes (MyCholesterol and Untargeted Marketing), each related to a specific data set of Personal Data and achieved by allowing access to perform a specific set of Actions. More precisely, the MyCholesterol purpose is achieved by performing AGGREGATE, DERIVE and QUERY actions; whereas the Untargeted Marketing purpose is achieved by performing COLLECT, QUERY and SEND actions.

At the time of subscribing to the myFitness application, Alice provided her personal data (i.e., e-mail, Age, Gender, and Blood Cholesterol) and gave her consent to process her e-mail and Age for Untargeted Marketing purpose, and her Blood Cholesterol for MyCholesterol purpose. Additionally, Alice withhold her consent to share her personal data with a third-party company named xxx-HealthOrg company. In turn, myFitness gave to Alice controller's contacts that include: piiController, orgName, address, e-mail, and phone number.

According to GRADUATION methodology, the application of the first two activities (see Figure 1) involves:

- **Model Derivation (Step ①)**: starting from the GDPR text, among the different proposals, the Privacy Ontology (PrOnto) [20, 21] ontology representation of the GDPR is used for deriving the GDPR's entities useful for modeling GDPR-based ABAC policy. In particular we considered:
  1. *Data* that is the object of the GDPR and it is target of its protection. Data can be: Personal Data, non-personal data, anonymized data and pseudonymised data;
  2. *Agents and Roles* such as data subject, controller, processor, supervisory authority and the new introduced figure the Data Protection Officer (DPO), as well as third-party;
  3. *Processing activities* expressed as a set of actions such as delete, transmit and store;
  4. *Purposes and legal bases* such as the consent; and finally,

5. *Legal rules* such as right, obligation, permission and prohibition.

According to the methodology presented in [6], an example of an abstract representation of a GDPR-based ABAC policy model is reported below:

((Subject = Controller OR DataSubject) ∧ (Resource = PersonalData) ∧ (Action = processing) ∧ (Action.purpose = PersonalData.purpose) ∧ (PersonalData.purpose.consent = YES)) ⟹ (Authorization = Permit)

– **Model Customization (Step ②)**: the GDPR-based ABAC policy model is then analyzed to identify main legal concepts and associate to each of them the proper input domain. In particular, based on the above scenario, a possible access control policy can be derived as reported here below. The policy allows a lawfulness of processing of personal data related to Alice in case of subscription to the myFitness specific service for two different purposes.

LawfulnessOfProcessingPolicy:

R1: permission(Controller=myFitness, DataSubject=Alice PersonalData=Blood Cholesterol, purpose=MyCholesterol, Action=DERIVE Consent=TRUE)
R2: permission(Controller=myFitness, DataSubject=Alice, PersonalData=EmailDS, purpose=UntargetedMarketing, Action=SEND Consent=TRUE)

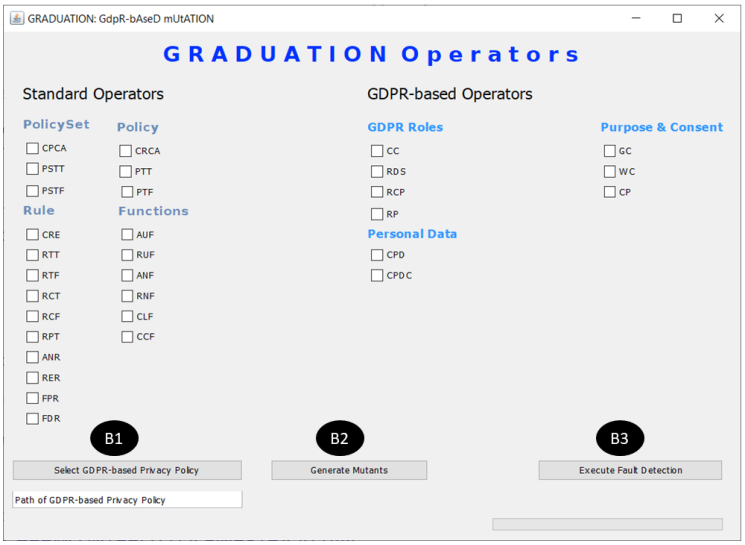

Fig. 3. GRADUATION Main GUI.

The **GDPR-based Implementation Parsing (Step ③)** is being performed through the application of the implementation of Module Ⓑ (see Figure 2). The end-user interaction is managed through an User Interface (UI) as

depicted in Figure 3. Through this interface, the end-user can select the GDPR-based policy (button **B1** in Figure 3) and starts its parsing. This access control policy, representing the gold implementation, is then analyzed for identifying the data useful for deriving its mutated versions. In Table 1, the result of this activity is represented. In particular, the table reports the set of GDPR-based entities (column *GDPR Entity*), their classification (column *Category*), their names (column *Name*) and related values (column *Value*). In the following, we describe the application of the GDPR-based mutation operators by considering the above policy named *LawfulnessOfProcessingPolicy*.

GDPR Entity	**Category**	**Name**	**Value**
Controller	Agent	orgName	myFitness
Controller	Biodata	piiController	myFitnessID
Controller	Biodata	address	-
Controller	Biodata	e-mailC	-
Controller	Biodata	phone number	-
Third-party	Agent	orgName	xxx-HealthOrg
Data Subject	Agent	DSName	Alice
Personal Data	Biodata	Age	-
Personal Data	Biodata	Gender	-
Personal Data	Biodata	Blood Cholesterol	-
Personal Data	Biodata	e-mailDS	-
Purpose	-	Purpose	MyCholesterol
Purpose	-	Purpose	Untargeted Marketing
Processing	-	Action	AGGREGATE
Processing	-	Action	DERIVE
Processing	-	Action	QUERY
Processing	-	Action	COLLECT
Processing	-	Action	SEND

Table 1. The GDPR Entities Extracted from the Model.

The **Mutation Operators Application (Step ④)** is performed through Module Ⓒ. In particular, by means of the User Interface (UI) (see Figure 3) the end-user can select the GDPR-based mutation operators and the standard ones, and apply them to the selected policy (button **B2**).

In the following, some examples of mutants related to *LawfulnessOfProcessingPolicy* are reported. In particular, in bold-italics text we report the name of the applied Mutation Operator, whereas in bold-blue we highlight the applied mutation operators within **R1** and **R2** rules.

Finally, the end-user can execute the policy mutants against a given test suite (button **B3** of the User Interface (UI) of Figure 3).

WC MUTANT
LawfulnessOfProcessingPolicy-WC1:

R1: permission(Controller=myFitness, DataSubject=Alice PersonalData=Blood Cholesterol, purpose=MyCholesterol, Action=DERIVE **Consent=FALSE**)
R2: permission(Controller=myFitness, DataSubject=Alice, PersonalData=e-mailDS, purpose=UntargetedMarketing, Action=SEND Consent=YES)

CP MUTANT
LawfulnessOfProcessingPolicy-CP2:

R1: permission(Controller=myFitness, DataSubject=Alice PersonalData=Blood Cholesterol, **purpose=UntargetedMarketing**, Action=DERIVE Consent=TRUE)
R2: permission(Controller=myFitness, DataSubject=Alice, PersonalData=e-mailDS, purpose=UntargetedMarketing, Action=SEND Consent=TRUE)

CPD MUTANT
LawfulnessOfProcessingPolicy-CPD:

R1: permission(Controller=myFitness, DataSubject=Alice **PersonalData=AGE**, purpose=MyCholesterol, Action=DERIVE Consent=TRUE)
R2: permission(Controller=myFitness, DataSubject=Alice, PersonalData=e-mailDS, purpose=UntargetedMarketing, Action=SEND Consent=YES)

In particular, this step involves the selection of the set of AC requests to be evaluated; the execution of the policy and the derived mutants against test suite; and the evaluation of which mutants (both standard and GDPR-based) have been killed by the application of the selected test suite. It is out of the scope of this work discussing the results of this step because it strictly depends on the test generation strategy or the test suite to be evaluated. The aim of this work is therefore to present a mutation strategy targeting the GDPR's peculiarities, and provides specific mutation operators based on the GDPR.

8 Conclusions

In this paper, we introduced GRADUATION, a comprehensive methodology for defining and applying mutation operators specifically conceived in the context of the GDPR. The methodology and the proposed mutation operators have been voluntarily conceived independent from any modeling language, used for formally represent the GDPR. Although grounded in a domain-related implementation (i.e., the GDPR), GRADUATION yields a more general spectrum, since it can be applied to different data protection regulations, and more in general to any legal text that implicitly contains, or suggests, data protection requirements. The applicability of GRADUATION has been exemplified in the context of ABAC domain. Thus, the ABAC-based GRADUATION implementation has been used to generate mutated versions of GDPR-based ABAC Privacy policies. Currently, we are working to extend the GDPR-based mutation operators set so as to cover other GDPR's demands as well as to improve its validation with real case studies. Ongoing work includes also the specialization of GRADUATION methodology considering other formalisms and languages such as UML and Semantic Web Technologies.

Acknowledgment

This work is partially supported by the project BIECO H2020 Grant Agreement No. 952702, and by CyberSec4Europe H2020 Grant Agreement No. 830929.

References

1. Barsocchi, P., Calabrò, A., Crivello, A., Daoudagh, S., Furfari, F., Girolami, M., Marchetti, E.: A privacy-by-design architecture for indoor localization systems. In: Proceedings of QUATIC 2020, Faro, Portugal, September 9-11, 2020. Communications in Computer and Information Science, vol. 1266, pp. 358–366. Springer (2020)
2. Bartolini, C.: Software testing techniques revisited for owl ontologies. In: International Conference on Model-Driven Engineering and Software Development. pp. 132–153. Springer (2016)
3. Bartolini, C., Calabrò, A., Marchetti, E.: Enhancing business process modelling with data protection compliance: An ontology-based proposal. In: Proceedings of the 5th International Conference on Information Systems Security and Privacy, ICISSP 2019, Prague, Czech Republic, February 23-25, 2019. pp. 421–428 (2019)
4. Bartolini, C., Calabrò, A., Marchetti, E.: GDPR and business processes: an effective solution. In: Proceedings of the 2nd International Conference on Applications of Intelligent Systems, APPIS 2019, Las Palmas de Gran Canaria, Spain, January 07-09, 2019. pp. 7:1–7:5 (2019)
5. Bartolini, C., Daoudagh, S., Lenzini, G., Marchetti, E.: Gdpr-based user stories in the access control perspective. In: Quality of Information and Communications Technology. pp. 3–17. Springer International Publishing, Cham (2019)
6. Bartolini., C., Daoudagh, S., Lenzini., G., Marchetti., E.: Towards a lawful authorized access: A preliminary gdpr-based authorized access. In: Proceedings of the 14th International Conference on Software Technologies - Volume 1: ICSOFT,. pp. 331–338. INSTICC, SciTePress (2019)
7. Basin, D., Debois, S., Hildebrandt, T.: On purpose and by necessity. In: Proceedings of the Twenty-Second International Conference on Financial Cryptography and Data Security (FC) (February 2018)
8. Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti., E.: Xacmut: Xacml 2.0 mutants generator. In: Proc. of 8th International Workshop on Mutation Analysis. pp. 28–33 (2013)
9. Daoudagh, S., Lonetti, F., Marchetti, E.: Assessment of access control systems using mutation testing. In: Proceedings of the First International Workshop on TEchnical and LEgal aspects of data pRIvacy. pp. 8–13. IEEE Press (2015)
10. Daoudagh, S., Marchetti, E., Savarino, V., Bernardo, R.D., Alessi, M.: How to improve the GDPR compliance through consent management and access control. In: Proceedings of the 7th International Conference on Information Systems Security and Privacy, ICISSP 2021, Online Streaming, February 11-13, 2021. pp. 534–541. SCITEPRESS (2021)
11. Davari, M., Bertino, E.: Access control model extensions to support data privacy protection based on gdpr. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 4017–4024 (2019)

12. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union **L119**, 1–88 (May 2016), http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC
13. Hu, C.T., Ferraiolo, D.F., Kuhn, D.R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K.: Guide to attribute based access control (abac) definition and considerations [includes updates as of 02-25-2019]. Tech. rep. (2019)
14. Jin, X., Krishnan, R., Sandhu, R.: A unified attribute-based access control model covering dac, mac and rbac. In: Data and Applications Security and Privacy XXVI. pp. 41–55. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
15. Khatibsyarbini, M., Isa, M.A., Jawawi, D.N., Tumeng, R.: Test case prioritization approaches in regression testing: A systematic literature review. Information and Software Technology **93**, 74 – 93 (2018)
16. Libal, T., Steen, A.: Towards an executable methodology for the formalization of legal texts. In: International Conference on Logic and Argumentation. pp. 151–165. Springer (2020)
17. Martin, E., Xie, T.: A fault model and mutation testing of access control policies. In: Proc. of WWW. pp. 667–676 (2007)
18. Mouelhi, T., Fleurey, F., Baudry, B.: A generic metamodel for security policies mutation. In: Proc. of ICSTW. pp. 278–286 (2008)
19. OASIS: eXtensible Access Control Markup Language (XACML) Version 3.0. http:docs.oasis-open.orgxacml3.0xacml-3.0-core-spec-os-en.html (2013)
20. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Legal ontology for modelling GDPR concepts and norms. In: Legal Knowledge and Information Systems - JURIX 2018: The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018. pp. 91–100 (2018)
21. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal reasoning. In: Electronic Government and the Information Systems Perspective - 7th International Conference, EGOVIS 2018, Regensburg, Germany, September 3-5, 2018, Proceedings. pp. 139–152 (2018)
22. Papadakis, M., Kintis, M., Zhang, J., Jia, Y., Le Traon, Y., Harman, M.: Mutation testing advances: an analysis and survey. In: Advances in Computers, vol. 112, pp. 275–378. Elsevier (2019)
23. Ramadan, Q., Salnitriy, M., Strüber, D., Jürjens, J., Giorgini, P.: From secure business process modeling to design-level security verification. In: Proceedings of MODELS 2017. pp. 123–133. IEEE (September 2017)
24. Ranise, S., Siswantoro, H.: Automated legal compliance checking by security policy analysis. In: Proceedings of SAFECOMP 2017. LNCS, vol. 10489, pp. 361–372. Springer (2017)
25. Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., Lenzini, G.: Formalizing gdpr provisions in reified i/o logic: the dapreco knowledge base. Journal of Logic, Language and Information **29**(4), 401–449 (2020)