Automating GDPR Compliance Verification for Cloud-hosted Services

Masoud Barati, George Theodorakopoulos, Omer Rana
School of Computer Science & Informatics
Cardiff University
Cardiff, UK
{BaratiM,TheodorakopoulosG,RanaOF}@cardiff.ac.uk

***Abstract*—Cloud-hosted business processes require access to customer data to complete a transaction, to improve a customer's on-line experience or provide useful product recommendations. However, privacy concerns associated with the use of this data have led to legal regulations that impose restrictions on how such data is requested or processed by an on-line service, with large penalties for violating these restrictions, e.g. the European General Data Protection Regulation (GDPR). We propose a framework for helping cloud-hosted services automate GDPR compliance checking. The framework comprises three steps: represent data flow in business processes with an appropriate abstraction (timed transition systems), formalise GDPR rules and obligations and incorporate them into the same abstraction, and implement the abstraction in a model checking tool (Uppaal) in order to automatically verify compliance of business process activities with GDPR. We demonstrate the approach using a cloud-based purchase order system.**

***Index Terms*—timed automaton, business process models, verification, data privacy**

I. Introduction

Modern businesses that make use of cloud-hosted services ingest and process a range of data about their customers: names, contact and shipping details, payment and billing information, demographic and past purchase information. Some of this information is crucial and necessary for the business to carry out the service that the customer requests (e.g. fulfill a purchase order). Other information is useful for improving the general customer experience (e.g. customisation/localisation of the website according to customer preferences), and for enhancing the business relationship with a customer (e.g. recommend new or complementary products and services; send notifications for discounts or special offers). Customer profiling services of this kind are now widely used, and form a key component of many cloud hosted e-commerce platforms.

Although useful for a cloud service provider, this personal information can also be potentially very sensitive for a customer. For instance, financial information can be particularly sensitive, as a data leak associated with credit card information can directly lead to financial fraud. Personal information may also used to impersonate a customer (resulting in identity fraud) and e.g. make a fraudulent purchase in their name, or even open a bank account or make a loan application. Recent data breaches, e.g. Equifax [1], have caused widespread concern about the privacy of individuals.

To address the privacy concerns of citizens, the European Union (EU) introduced the General Data Protection Regulation (GDPR), which imposes a series of demands on businesses that handle EU citizen data. Potential penalties can be up to 20 million euros, or 4% of the annual global turnover – whichever is higher, in case of non-compliance. As a result, it is important for many businesses to modify their data-handling processes in order to comply with this new regulation. This adaptation, if done manually, is a difficult and error-prone process that needs significant attention to detail to maintain on an ongoing basis.

Various researchers have considered regulatory compliance for on-line business processes [2]–[4], but have not made it possible to automate compliance checking of activities carried out by processing units on user data. We propose a framework that can automate GDPR compliance checking. Our approach is to formalise GDPR rules using timed transition systems that abstract business process models. We represent a business process model for a cloud-based order system, where activities process specific customer data. We extend the formulation of timed automaton to support GDPR compliance checking of these activities. After the transformation of the business process model into a timed automaton, we implement the automaton in Uppaal and propose temporal logic formulas to verify a set of GDPR rules over the system.

The rest of the paper is organized as follows. Section II mentions related literature to provide a context for this work. Section III represents a cloud-based order scenario and Section IV describes GDPR rules associated with this scenario. Section V extends the formulation of timed automata for checking GDPR compliance in Uppaal. Finally, we conclude our paper in Section VI.

II. Literature Review

Using a particular case study, Saeki et al. [2] map business processes into transition systems to provide automatic regulatory compliance checking. A method for preserving privacy in a structured representation (using XML) and information brokering is proposed in [3] that uses an innovative automaton segmentation scheme. The method proposed an integrated security enforcement and query forwarding technique to preserve

This work has been funded by EPSRC award: EP/R033439/1

system-wide privacy. An approach for analysing regulatory compliance of software requirements with the aid of querying a production rule model was also introduced. In [4], an automated technique for reasoning about the semantics of requirements specification documents was proposed through which a set of logical formulas were generated from privacy and regulatory statements. An approach specifying the purpose (i.e. the intended outcome) associated with a business process was proposed in [5] to illustrate how formal models of interprocess interactions can be used to track GDPR regulations within a system. In [6], a model-based technique to enable data-aware compliance checking of business processes is proposed. The model showed how state space explosion can be avoided by conducting compliance checking for an abstract business process and abstract compliance rules. However, these approaches are limited in their scope and potential use. For instance, these approaches cannot be directly used to provide verification of storage, profiling and transfer of personal data in accordance with GDPR obligations.

The verification of IoT-based applications under GDPR rules was presented in [7], where several GDPR rules were encoded in smart contracts to automatically protect IoT user data. In [8], a privacy-aware cloud architecture was proposed to enhance transparency, and enable the tracking of providers who accessed user data. The architecture took advantages of both GDPR rules and a blockchain network to verify GDPR compliance. However, these contributions have not formally examined the verification of GDPR rules at design time, and can only be used once a transaction has been completed. Moreover, they have not been used to verify user data retention period on the storage system of a cloud provider (another key requirement of GDPR compliance).

III. A CLOUD-BASED PURCHASE ORDER SYSTEM

We consider a cloud-based purchase order scenario to demonstrate how the proposed approach can be used. In this scenario, a customer requests goods through a Web portal, pays with a credit card, and receives their order via an online delivery system. The order system also provides targeted marketing, such as online advertisements, to send new offers to their subscribed customers. The assumption is that each part of the overall system is handled by a different cloud provider: one for registering customer information, one for managing orders, and one for sending advertisements. Fig. 1 illustrates the core business processes of such a system in Business Process Model and Notation (BPMN) representation, concentrating only on personal data collected and processed. The business processes in this scenario are as follows:

Customer registration: A cloud customer signs and subscribes through a Web portal for online purchase. Through this business process, customers should provide their email, postal address and credit card information. The process stores the customer data and also sends a copy to other providers involved in the transaction.

Online purchase: Registered customers can select products from a Web catalogue and pay using a credit card already recorded during the customer registration process. The business process issues order receipts, stores them locally and submits such receipts to the targeted marketing provider.

Targeted marketing: The email or surface mail address of a customer is used to send targeted advertisements based on the purchase history of a customer. This process creates an individual profile for each customer from their recorded orders and keeps the profiled data on local storage.

Hence activities on customer data handled by the aforementioned business processes may be classified into *access*, *store*, *transfer* and *profile*. For instance, the access activity appears in each process. Similarly, the store activity is executed by all business processes, which keep customer information (name, address, etc.), order information (purchased goods, buyer identification, etc.) and profile data (customer age, purchased goods by customer, etc.) in their local storage, respectively. The transfer activity appears in both register customer and online purchase operations through which collected data is sent to other processes. Finally, the targeted marketing is the only process running the profile activity. According to the main roles defined in GDPR, the provider handling registration process in Fig. 1 can be classified as both data controller and processor. Its role is a controller when it collects data and delivers them to other providers. It also has a processor role when recording data in its local storage. Likewise, the provider running the purchase process is a data processor during order creation activities and has a joint controller role when order information is submitted for the marketing process. The provider managing the advertisement procedures can play the role of data processor, since it profiles and stores customer data. Cloud customers are data subjects who provide personal data such as name, age, address, credit card information and identification number.

Using BPMN notation as illustrated in Fig 1 three pools are identified, one for each process. Each pool has a number of activities within a swimlane, some of which are handled by external entities such as a human operator while others are automated. The boxes with dark envelopes show data transfer. The activities marked with script icons involve customer profiling. The solid arrows between activities denote their sequence in a design pattern. Each activity may use or produce data recorded in databases, demonstrated by dashed arrows. Databases permit data to be shared between business processes.

IV. GDPR OBLIGATIONS

The GDPR legislation proposes multiple rules and obligations on activities carried out by data controllers/processors (or actors for short). For example, the regulations associated with activities depicted in Fig. 1 are described below.

Access: Article 32(1)(a) of GDPR requires actors who access personal data to employ encryption for preventing unauthorized data access.

Store: Article 17 of GDPR requires actors who store personal data to provide users with the capability to erase their personal data at any time (referred to as the “Right to be Forgotten”).

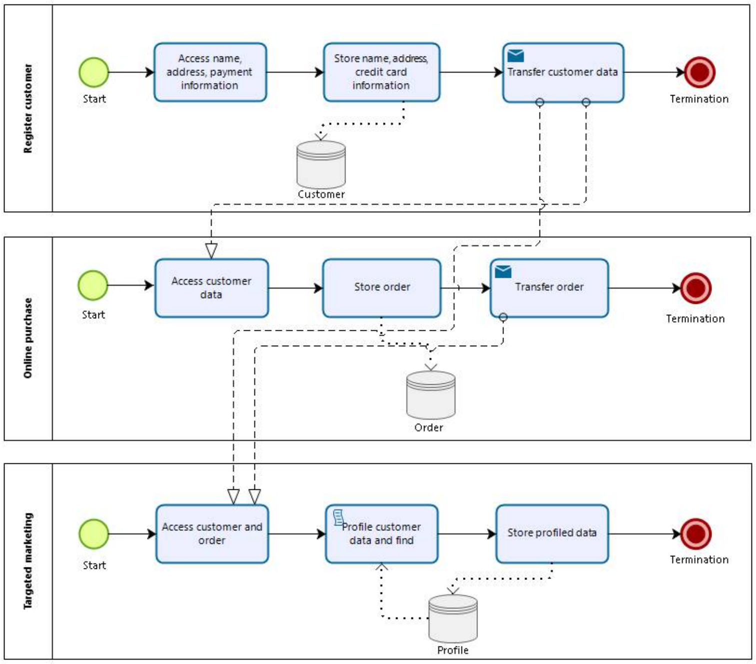


Fig. 1. Cloud-based purchase order system

Moreover, Article 5(1)(e) does not allow actors to store personal data longer than the time necessary for data processing. **Profiling:** Article 22 of GDPR states that any profiling operation on users who are under 18 years old is not permitted. **Transfer:** Articles 44–47 of GDPR restricts actors (responsible for handling user data) on transferring personal data only inside Europe or countries holding Binding Corporate Rules (BCR) certification. BCR are internal rules (e.g. code of conduct) which are adopted by a community of multinational companies that want to transfer personal data internationally across various jurisdictions [9].

Before the execution of such activities on personal data, GDPR enforces actors to obtain user consent (Recital (32), (43) of GDPR). Moreover, according to Article 32(1)(a) of GDPR, actors should implement measures such as encryption to ensure the protection of personal data during the execution of the aforementioned activities.

GDPR rules force actors to satisfy a number of conditions prior to executing an activity on customer data. For instance, storing data is subject to (a) obtaining customer consent; (b) the encryption of data; (c) the option to erase data at any time; (d) the retention period being less than the total time needed for all data processes. The verification of these GDPR obligations in a formal or automatic way has become a challenge. We consider the use of finite state machines as a means to address this challenge.

V. From business process to timed automaton

GDPR mainly concentrates on the key purpose of data processing and encourages system developers to show such *purpose* in a more open and transparent way. Each business process belonging to a process collection model can express this purpose (of data processing) [5]. By designing a business process collection model, actors determine processing activities that should be executed on user data during the service execution life cycle. However, a formal abstract model is required to facilitate the verification of GDPR obligations over such activities. In order to undertake verification with the aid of model checking tools, the process collection models should first be abstracted by finite state machines. GDPR compliance of this derived automata, representing the purposes of data processing by actors, can be verified by using a set of temporal logic formulas.

Actors sometimes determine deadlines or time constraints for the execution of their activities, particularly when they want to notify customers about the processing time or retention period of data in their local storage. Hence, timed automata can be a useful means for supporting the time constraints associated with the completion of activities. One of the elements of a timed automaton is a set of timed variables – referred to as "clocks" C, which are non-negative real-valued variables. A conjunctive formula of terms in the form of $x \sim c$ or $y - x \sim c$, where $x, y \in C$, $c \in \mathbb{N}$, and $\sim \in \{\leq, <, =, >, \geq\}$, is used to express a clock/time constraint [10].

The GDPR rules associated with an activity can be expressed using a number of Boolean variables, confirmed by both a user and the data processing actor. For example, GDPR issues associated with *profile* activity are user consent, data encryption, and customer age. These issues, denoted here as *consent*, *encrypt*, and *is-adult* with `true` or `false` values, can form the attributes of the activity. Formally, a set of attributes is denoted by $Att = \{att_1, \cdots, att_l\}$ such that each attribute att_i can be associated with a domain of values

$V_i = dom(att_i)$. A valuation for a set of attributes Att is a function $v : Att \rightarrow V_1 \cup \cdots \cup V_l$, assigning a value $v(att_i) \in V_i$ to every attribute $att_i \in Att$. Defining such attributes extends the formulation of a timed automaton as follows.

Definition 1. A timed automaton that supports attributes associated with activities is a tuple: $\langle \mathcal{A}, Att, \gamma, Q, q_0, F, C, \eta, H \rangle$, where $\mathcal{A}$ is a finite set of activities; Att is a set of attributes; $\gamma : \mathcal{A} \rightarrow 2^{Att}$ is a function that can assign a subset of attributes to an activity; Q is a finite set of states; $q_0 \in Q$ is the initial state; $F \subseteq Q$ is the set of final states; C is a finite set of clocks; $\eta \subseteq Q \times \mathcal{A} \times G(C) \times 2^C \times Q$ is the transition relation, with $G(C)$ the set of constraints over C; and $H : Q \rightarrow G(C)$ is a function, which assigns an invariant to every state. The latter indicates the time that may be spent in a state (also called location). The tuple $\langle q, \alpha, g, D, q' \rangle$ is a transition from q to q' on activity α with the clock constraint g. The clocks that belong to $D \subseteq C$ are reset to zero when the transition is taken.

Example 1. Consider that in the business process collection model in Fig. 1, actors determine time constraints for the storage of personal data between 45 and 60 minutes, between 15 and 20 minutes, and between 30 and 40 minutes in the *register, purchase*, and *targeted marketing* processes, respectively. The activities of the business process model appear in the transitions of the finite state machine and each state shows the status of the business process after the execution of an activity. For better illustration of activities, they appear with the name of their associated process. The *access* activity involves *consent* and *encrypt* attributes. The *store* activity has *consent*, *encrypt* and *erase* attributes, where the latter shows the capability of erasing data by the actor at any time. Moreover, the activity has a clock x to represent the time interval over which data can be kept in storage. The *transfer* activity contains *consent*, *encrypt*, *EU*, and *BCR* attributes. The two last ones show whether the data is transferred outside the European Union and whether the data receiver supports BCR or not. Finally, the $profile$ activity has *consent*, *encrypt*, and *is-adult* attributes. State t_1 is an initial state and the states t_4, t_7, and t_{10} denote final states, where the business processes are successfully terminated.

A. Implementation of proposed timed automaton in Uppaal

The finite state machine proposed for the abstraction of a business process can be implemented via the model checking tools that support time constraints. One such tools is Uppaal, providing an environment for modeling, simulation and verification of real-time systems represented by timed finite state machines [11]. A comparison between Uppaal and other timed model checking tools such as Specification Description Language (SDL) and Timed Petri Nets (TPN) showed that Uppaal has a better performance in terms of time and memory used for verifying timed transition systems [12]. Uppaal provides the following: (i) a modeling formalism through which a real-time system can be designed and implemented using timed automata; (ii) a simulator that provides a stepwise and random execution of the real-time system; and (iii) a model checker that verifies the system under several properties (i.e. safety, liveness and reachability).

The timed automaton modeling the abstraction of the business process collection model can have its local clock. A clock is declared as "`clock x`" in the declaration part of the template related to a timed automaton. Let the interval $[l_1, l_2]$, where l_1 and l_2 are integers represent time interval for performing an activity using a timed automaton with clock x. Once the activity is executed, the automaton reaches a new state t_k. In order to implement such a time interval in Uppaal, a time constraint $x \geq l_1$ should be associated with the transition for the activity and an invariant $x \leq l_2$ associated with state t_k in the automaton. Figure 3 illustrates the implementation of the finite state machine for the cloud-based purchase order system in the Uppaal simulator, where transitions are followed sequentially. A clock x appears on the transitions and states associated with both store and transfer data activities and states $t3$ and $t10$ show two invariants defining the upper bounds of clock constraints. The Boolean variable *consent* is set by the data subject and the rest are set by the actor. For instance, if the actor does not support the encryption of personal data for an activity, the value of *encrypt* is set to 0 on the activity transition. In Uppaal, such values are assigned in the *update* part associated with a transition.

B. Verification of proposed timed automaton in Uppaal

A set of TCTL formulas can be verified in Uppaal. Let ϕ be a state formula. A safety property can be written as a formula of the form $A[\,]\ \phi$ or $E[\,]\ \phi$. The former states that ϕ should be true for all reachable states, while the latter is used to check whether there exists a path such that ϕ is always true. The safety property $A[\,]\ \phi$ is used for verification, since it is stronger than liveness and reachability properties [11].

Given the aforementioned GDPR obligations expressed in Section IV, the following verification can be performed on actors with respect to their activities over user data.

Access activity: Let t_k be a reachable state just after the execution of access operation α in a timed automaton p. To check for GDPR compliance of p with respect to the rules legislated for α, the following formula must be satisfied for all $\langle t_{k-1}, \alpha, t_k \rangle$ in p:

$$\texttt{A[]}\ p.\alpha\ \texttt{and}\ p.t_k\ \texttt{imply}\ p.encrypt{==}1\ \texttt{and}\ p.consent{==}1 \tag{1}$$

Informally, the formula states that execution of the access activity is subject to data encryption and user consent, otherwise it is not compliant with GDPR.

Store activity: Let t_k be a reachable state just after the execution of store operation α in a timed automaton p and x is a clock showing the storage time. To check for GDPR compliance of p with respect to the rules legislated for α, the following formula must be satisfied for all $\langle t_{k-1}, \alpha, t_k \rangle$ in p:

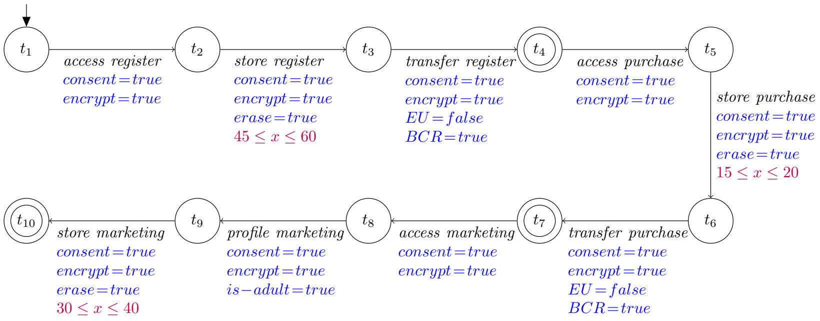


Fig. 2. State transitions for cloud-based purchase order system

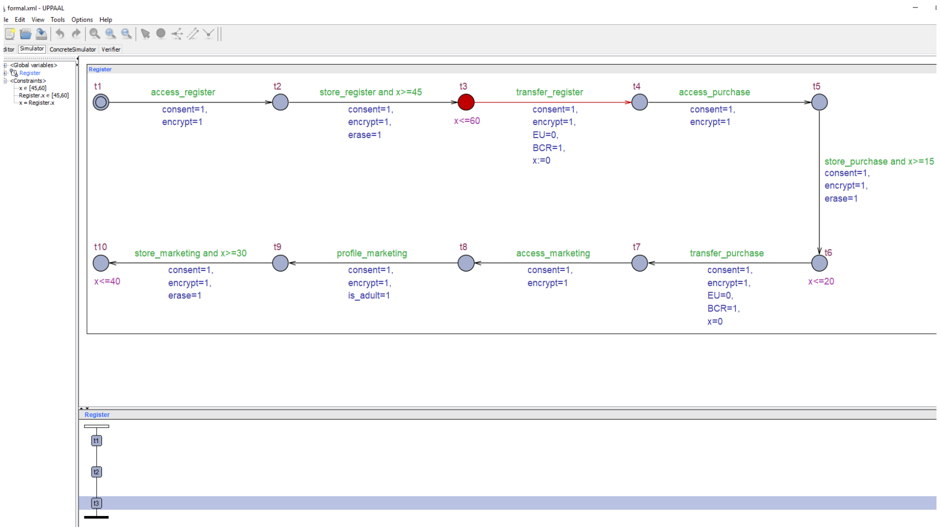


Fig. 3. Implementation of register customer transition system in Uppaal

$$
\begin{aligned}
&\texttt{A[]}\ \ p.\alpha\ \ \texttt{and}\ \ p.t_k\ \texttt{imply}\\
&\quad p.encrypt{=}{=}1\ \ \texttt{and}\ \ p.consent{=}{=}1\ \ \texttt{and}\\
&\quad p.erase{=}{=}1\ \ \texttt{and}\ \ p.x \le l
\end{aligned}
\tag{2}
$$

where l is an upper bound, being acceptable for retention period according to GDPR. Basically, retention period is less than or equal to the time required for executing all processing activities on user data.

Profile activity: Let t_k be a reachable state just after the execution of profile operation α in a timed automaton p. To check for GDPR compliance of p with respect to the rules legislated for α, the following formula must be satisfied for all $\langle t_{k-1}, \alpha, t_k \rangle$ in p:

$$
\begin{aligned}
&\texttt{A[]}\ \ p.\alpha\ \ \texttt{and}\ \ p.t_k\ \texttt{imply}\\
&\quad p.encrypt{=}{=}1\ \ \texttt{and}\ \ p.consent{=}{=}1\ \ \texttt{and}\\
&\quad p.is\text{-}adult{=}{=}1
\end{aligned}
\tag{3}
$$

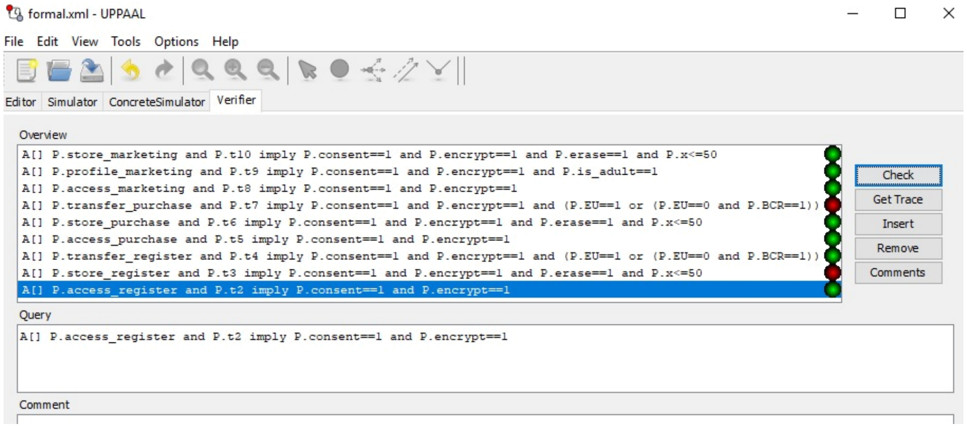
The formula means that the execution of profile activity is only allowed for adults (i.e. age>18) and subject to data encryption and user consent.

Transfer activity: Let t_k be a reachable state just after the execution of transfer operation α in a timed automaton p. To check GDPR compliance of p with respect to the rules legislated for α, the following formula must be satisfied for all $\langle t_{k-1}, \alpha, t_k \rangle$ in p:

$$
\begin{aligned}
&\texttt{A[]}\ \ p.\alpha\ \ \texttt{and}\ \ p.t_k\ \texttt{imply}\\
&\quad p.encrypt{=}{=}1\ \ \texttt{and}\ \ p.consent{=}{=}1\ \ \texttt{and}\\
&\quad (p.EU{=}{=}1\ \ \texttt{or}\ \ (p.EU{=}{=}0\ \ \texttt{and}\ \ p.BCR{=}{=}1))
\end{aligned}
\tag{4}
$$

The formula states that the execution of transfer activity requires encryption and consent of user. When the transfer is outside Europe, the receiver of user data must follow BCR.

Example 2. Consider that customer data (within the cloud-based purchase order system) must be maintained on the storage system of the cloud provider for less than (or equal to) 50 minutes. Moreover, the actor undertaking the targeted marketing process is outside Europe and has not received BCR certification. Given these assumptions, we verify the timed automaton depicted in Fig. 3 in accordance with the proposed formulas (Eq. 1 to Eq. 4). Figure 4 shows verification operations undertaken by the model checker in Uppaal. As seen, the verification of two formulas are not satisfied (marked



Overview

```
A[] P.store_marketing and P.t10 imply P.consent==1 and P.encrypt==1 and P.erase==1 and P.x<=50  🟢
A[] P.profile_marketing and P.t9 imply P.consent==1 and P.encrypt==1 and P.is_adult==1  🟢
A[] P.access_marketing and P.t8 imply P.consent==1 and P.encrypt==1  🟢
A[] P.transfer_purchase and P.t7 imply P.consent==1 and P.encrypt==1 and (P.EU==1 or (P.EU==0 and P.BCR==1))  🔴
A[] P.store_purchase and P.t6 imply P.consent==1 and P.encrypt==1 and P.erase==1 and P.x<=50  🟢
A[] P.access_purchase and P.t5 imply P.consent==1 and P.encrypt==1  🟢
A[] P.transfer_register and P.t4 imply P.consent==1 and P.encrypt==1 and (P.EU==1 or (P.EU==0 and P.BCR==1))  🟢
A[] P.store_register and P.t3 imply P.consent==1 and P.encrypt==1 and P.erase==1 and P.x<=50  🔴
A[] P.access_register and P.t2 imply P.consent==1 and P.encrypt==1  🟢
```

Query

```
A[] P.access_register and P.t2 imply P.consent==1 and P.encrypt==1
```

Comment

Fig. 4. Verification in Uppaal

in red).[1]

(1) `A[] P.store_register and P.t3 imply`
`P.consent==1 and P.encrypt==1 and`
`P.erase==1 and P.x<=50`

where `P` is the name of the timed automaton. As the actor undertaking the register customer process set retention period from 45–60 minutes, a violation is flagged, as retention period $\leq$ 50 minutes.

(2) `A[] P.transfer_purchase and P.t7 imply`
`P.consent==1 and P.encrypt==1 and`
`(P.EU==1 or (P.EU==0 and (P.BCR==1))`

The violation occurs in this instance due to the following: the actor undertaking targeted marketing does not hold a BCR certification.

VI. Conclusion

An automated mechanism for verifying GDPR rules for cloud-hosted services is proposed. These services are realised as a business process and require access to personal user data. An abstraction is used to capture the data flow associated with business process models using timed automata in order to facilitate the verification through available model checking tools. The formulation of timed automata was extended to include attributes associated with particular activities, each of which reflects a GDPR concern. A business process model for a cloud-based order system involving four typical activities (i.e. access, store, profiling, and transfer) is used to illustrate the approach. The case study illustrates how GDPR rules associated with particular activities can be verified using the approach. The timed automaton was implemented in Uppaal and several temporal logic formulas were verified on the system to detect possible GDPR violations. Our proposed logic formulas also enabled the verification of GDPR requirements dealing with the time constraints of processing activities. Future work will focus on generalising the presented approach and widening the range of activities carried out on personal data. Moreover, the integration of our approach with services that collect or process personal data in real time is another potential research direction.

[1] The assumption is that *consent* is reset after the transition associated with an activity on customer data.

References

[1] "Equifax Data Breach Settlement," [Online]. Available: https://www.ftc.gov/enforcement/cases-proceedings/refunds/equifax-data-breach-settlement, Accessed on: 15-04-2020.

[2] M. Saeki, H. Kaiya, and S. Hattori, "Detecting regulatory vulnerability in functional requirements specifications," in 4th Int. Conf. on Software & Data Technologies, Sofia, Bulgaria, 2009, pp. 105–114.

[3] F. Li, B. Luo, P. Liu, D. Lee, and C.H. Chu, "Automaton segmentation: A new approach to preserve privacy in XML information brokering," in 14th ACM Conf. on Computer & Comms. Security, Alexandria, VA, USA, 2007, pp. 508–518.

[4] R. Nakamura, Y. Negishi, S. Hayashi, and M. Saeki, "Terminology matching of requirements specification documents and regulations for compliance checking," in 8th Int. Workshop on Requirements Engineering and Law, Ottawa, ON, Canada, 2015, pp. 10–18.

[5] D. Basin, S. Debois, T. Hildebrandt, "On purpose and by necessity: Compliance under the GDPR," in Int. Conf. on Financial Cryptography and Data Security, Springer, Nieuwpoort, Curacao, 2018, pp. 20–37.

[6] D. Knuplesch, L.T. Ly, S. Rinderle-Ma, H. Pfeifer, and P. Dadam, "On enabling data-aware compliance checking of business process models," in Conceptual Modeling, Lecture Notes in Computer Science, J. Parsons, M. Saeki, P. Shoval, C. Woo, and Y. Wand (eds), Springer, Berlin, Heidelberg, vol. 6412, 2010.

[7] M. Barati, I. Petri, and O. F. Rana, "Developing GDPR compliant user data policies for Internet of things," in 12th IEEE/ACM International Conference on Utility and Cloud Computing, Auckland, New Zealand, 2019, pp. 133–141.

[8] M. Barati, O. Rana, G. Theodorakopoulos, and P. Burnap, "Privacy-aware cloud ecosystems and GDPR compliance," in 7th Int. Conf. on Future Internet of Things and Cloud, Istanbul, Turkey, 2019.

[9] M. Corrales, P. Jurcys, and G. Kousiouris, "Smart contracts and smart disclosure: coding a GDPR compliance framework," SSRN Electronic Journal, 2018.

[10] R. Alur and D.L. Dill, "A theory of timed automata," Theoretical Computer Science, vol. 126, pp. 183–235, 1994.

[11] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on Uppaal," in Formal Methods for the Design of Real-Time Systems, Lecture Notes in Computer Science, M. Bernardo, F. Corradini (eds), Springer, Vol. 3826, 2004, pp. 200–236.

[12] K. Godary-Dejean, I. Augé-Blum, and A. Mignotte, SDL and Timed Petri Nets versus UPPAAL for the validation of embedded architecture in automotive, in Int. Conf. on Forum on specification and Design Languages, Lille, France, 2004.