Institute of Architecture of Application Systems

TIMED PRIVACY-AWARE BUSINESS PROTOCOLS

KARIMA MOKHTARI-ASLAOUI, SALIMA BENBERNOU and SOROR SAHRI[1]
VASILIOS ANDRIKOPOULOS and FRANK LEYMANN[2]
MOHAND-SAID HACID[3]

1: LIPADE, Université Paris Descartes, France
2: Institute of Architecture of Application Systems, University of Stuttgart, Germany,
3: Université Lyon 1, France

BIBTEX:

```
@article{doi:10.1142/S0218843012500013,
author = {MOKHTARI-ASLAOUI, KARIMA and BENBERNOU, SALIMA and SAHRI, SOROR and
ANDRIKOPOULOS, VASILIOS and LEYMANN, FRANK and HACID, MOHAND-SAID},
title = {TIMED PRIVACY-AWARE BUSINESS PROTOCOLS},
journal = {International Journal of Cooperative Information Systems},
volume = {21},
number = {02},
pages = {85-109},
year = {2012},
doi = {10.1142/S0218843012500013},
URL = {http://www.worldscientific.com/doi/abs/10.1142/S0218843012500013},
eprint = {http://www.worldscientific.com/doi/pdf/10.1142/S0218843012500013}
}
```

Electronic version of an article published *Int. J. Coop. Info. Syst.* **21(2)**, 85-109 (2012). DOI: 10.1142/S0218843012500013 © World Scientific Publishing Company
http://www.worldscientific.com/doi/abs/10.1142/S0218843012500013

Universität Stuttgart
Germany

International Journal of Cooperative Information Systems
Vol. 21, No. 2 (2012) 85–109

DOI: 10.1142/S0218843012500013

World Scientific
www.worldscientific.com

TIMED PRIVACY-AWARE BUSINESS PROTOCOLS

KARIMA MOKHTARI-ASLAOUI*, SALIMA BENBERNOU†
and SOROR SAHRI‡

LIPADE, Université Paris Descartes, France

University of Oran, Computer Science Department, Algeria
**mokhtari.karima@univ-oran.dz*
†*salima.benbernou@parisdescartes.fr*
‡*soror.sahri@parisdescartes.fr*

VASILIOS ANDRIKOPOULOS§ and FRANK LEYMANN¶

Institute of Architecture of Application Systems
University of Stuttgart, Germany
§*vasilios.andrikopoulos@iaas.uni-stuttgart.de*
¶*frank.leymann@iaas.uni-stuttgart.de*

MOHAND-SAID HACID

Université de Lyon, Université Lyon 1,
LIRIS CNRS UMR 5205, France
mohand-said.hacid@univ-lyon1.fr

Web services privacy issues have been attracting more and more attention in the past years. Since the number of Web services-based business applications is increasing, the demands for privacy enhancing technologies for Web services will also be increasing in the future. In this paper, we investigate an extension of business protocols, i.e. the specification of which message exchange sequences are supported by the web service, in order to accommodate privacy aspects and time-related properties. For this purpose we introduce the notion of Timed Privacy-aware Business Protocols (TPBPs). We also discuss TPBP properties can be checked and we describe their verification process.

Keywords: Privacy; web services; business protocols; timed automata.

1. Introduction

Over the past years, there has been a widespread increase in the use of web-based services supporting different business applications. Such popularity is accompanied by an exponential amount of data exchanged and collected by interacting entities, and a number of pressing issues that should be resolved, especially the issues related to consumers' Personal Identifiable Information (PII). In fact, most of the time, web-based service providers require some personal information or financial information from their consumers. Such information might be used for a number of purposes, from access to their online services (authentication, authorization) to billing (accounting), to service maintenance and so on. Hence, today, the individuals

are becoming more and more concerned about the privacy of their personal data (see Refs. 1 and 2). In general, privacy policies describe an organization's data practices: What information they collect from individuals (e.g. consumers) and what (e.g. purposes) they do with it. To enable privacy protection for Web service consumers across multiple domains and services, the World Wide Web Consortium (W3C) published a document called Web Services Architecture (WSA) Requirements that define some specific privacy requirements for Web services as a future research topic. At this moment, there is still no standardized Web services privacy technology. That is, no current Web service modeling technology offers a simple way to state privacy requirements. Policies specified as rules under which conditions the private data can be collected have been proposed as a solution to this problem. We discuss such issues and present a formal model for privacy based on our previous work (see Refs. 3 and 4).

Services descriptions include the interface definitions on transport level defined in WSDL and the business protocol definitions. WS-BPEL[5] can be used to specify such protocols. However, in order to facilitate service development and to allow automated analysis of service descriptions a formal model is needed. In Refs. 6 and 7 for example, the authors developed an expressive business protocol model based on state machines. The proposed approach provides contributions to protocol analysis for functional aspects.

However, to the best of our knowledge no similar work has been done in the context of privacy. Moreover, one of the essential ingredients in the completeness of behavioral analysis is quantitative properties such as time. Little work has been conducted in this direction (see Refs. 8 and 9). Time-related properties are relevant in this setting particularly for privacy handling. Indeed, in many scenarios we expect that Web services satisfy some timed constraints regarding the collected personal data of a client. It is important to check whether the timed privacy properties are satisfied specifically for distributed business protocols.

For instance, to serve a request of a client, a current business application requires collecting an email address to send the invoice of a purchase order. It will also keep the private data during two months for future promotions. The latter activity is achieved by another business protocol called transversal business protocol. Afterwards, the collected data will be destroyed. The business protocols must comply with these timed constraints.

In this paper, we investigate privacy issues in the context of business protocols. Moreover, we emphasize the time related properties of privacy management. Indeed, to fulfill these objectives and address the privacy shortcoming discussed so far, we propose a formal framework for specifying and verifying time-related properties in privacy-aware Web service protocols. Our contributions are:

(i) We formally provide a privacy model that is suitable to represent and integrate the use of personal data in business protocols.
(ii) We define a *privacy-aware business protocol* integrating privacy in the conversation.

(iii) We formally propose the notion of *Timed Privacy-aware Business Protocols* (*TPBP*), an extension of privacy-aware business protocols that are suitable to represent timed privacy constraints.
(iv) We develop a timed properties verification model for TPBPs.

The paper is organized as follows: Section 2 provides the formal model for privacy to be used to annotate the business protocol, and introduces an illustrative example to highlight the different components. Section 3 discusses the integration of privacy concerns in business protocols by introducing privacy-aware business protocols. Section 4 presents TPBP, an extension of the privacy-aware business protocols, handling the different ingredients and temporal properties. The verification process of the proposed model is presented in Sec. 5. In Sec. 6, we survey works similar to ours in the industry and research communities, concerning privacy in IT World in general or specifically in web services. In this section, we also discuss the ongoing works concerning business protocols in the research community. Finally, Sec. 7 concludes and discusses future research directions.

2. Privacy Model

Because of the increasing popularity of Web services a number of pressing issues should be resolved, and especially the issues related to consumers' *PII*. Most of the time, Web-based service providers require some personal information or financial information from their consumers. Such information may be used for a number of purposes, from regular access to their online services (authentication, authorization) to billing (accounting), to service maintenance and so on.

Nowadays, people are becoming more and more concerned about the privacy of their personal data (see Refs. 1, 10–12). Privacy policies are used by Web services in order to ease the privacy concerns of their clients and to adhere to legislative measures, stating what they can do or cannot do with the personal information of their clients. The most significant effort currently underway to enable web site users to gain control over their private information is the Platform for Privacy Preferences Project (P3P),[13] developed by the W3C. P3P is designed to declare the web site operators intentions for the use of the data they collect about the user. Moreover, the Enterprise Privacy Language (EPAL) language[14] is designed to specify enterprise privacy concerns, focusing on access authorization to personal data. EPAL policies are expressed as rules which can be enforced through an implementation of a rule engine.

Our work does not make any assumption about the choice of the language used to specify the policies. They could be in one of the standard languages. What we would like to focus on is the encoding of such policies into Web services modeling business processes. Furthermore, privacy policies do not discuss the behavior of individual business applications within the organization that actually collect/analyze and distribute personal data. This makes the enforcement of the policies difficult. We introduce our model of privacy rules as an extension of the categories of rules

defined in the platform of privacy preferences P3P[1] for the purpose of capturing privacy abstractions while describing the behavior of Web services.

Motivating example

We consider a hotel booking service. The requester specifies the desired town in which she wants to make a reservation. Then, the service proposes to the client a list of hotels. Once the client has done her choice, she is requested to provide banking information (e.g. credit card number) to confirm the reservation. This data is very sensitive and the client hopes that it will be used only for the purpose she has specified. That means she wishes to be guaranteed privacy and secure use, despite the fact that the service may employ a number of third-party supporting services to actually realize her reservation banking services for example should retain her credit card number only for credit checking purposes. In addition, let us assume that the service offers also the choice of booking a car for the duration of the stay. In this case, the data of the client should be provided to the car booking service if and only if she wants to use this additional option.

Furthermore, suppose that due to a failure, one of these supporting services cannot achieve its task. Ideally, the failed service could be replaced transparently to the client. Since the client has provided her private data though the new service, it must:

- have the same functionalities as the old one, and
- ensure the same privacy level of the collected data that was guaranteed by the old one.

Existing approaches on replaceability[15] consider only functional properties. Nonfunctional aspects (e.g. privacy policies) which are equally important are not usually taken into account. We therefore believe that it is necessary to provide a formal model to represent privacy in business protocols. This will enable handling service replaceability in a way transparent to the client.

2.1. *Modeling privacy policies*

Before establishing an interaction between a client and a provider, the client specifies by means of rules called privacy preferences, the way the private data can be used by the provider. The provider specifies through rules called privacy policies, how it will use those private data. To establish a conversation between the client and the provider (in which the client provides its private data), the preferences of the client must be consistent with the policies of the provider. Since the service provider may also be consuming another Web service, (that is, acting in turn as a client), it should also specify its preferences. Thus, each Web service owns both preferences and policies. In the following, we present the ingredients that constitute a policy or respectively a preference.

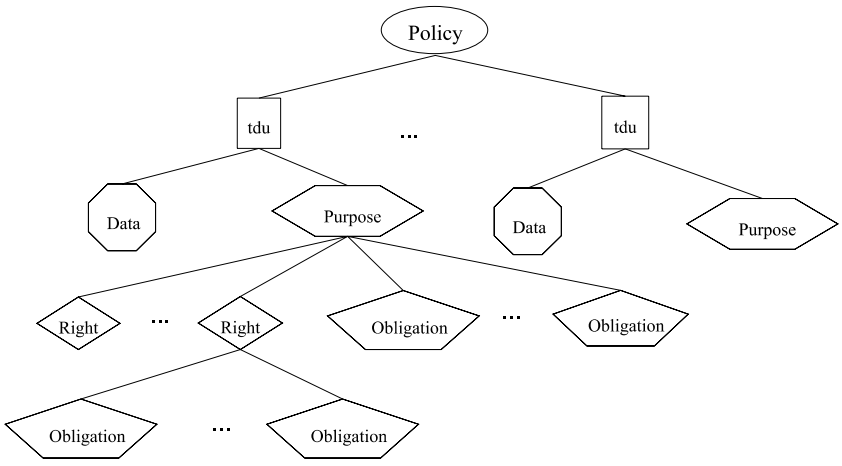


Fig. 1. A privacy policy model.

We define a privacy policy as a finite set of *Terms of Data Use* (*TDU*). A *tdu* $\in$ *TDU* consists of *private data* and *a purpose* for which the private data must be collected (see Fig. 1). A purpose is an action representing the need of the client executed by a given *entity*. Entities are services that can use the data to fulfill the request (*purpose*) of the client in a given *time frame*. Furthermore, the provider can require getting a choice to perform, or not, other actions called *Rights*. Since the fulfillment of purposes and rights involves the use of the client private data, the provider must guarantee their security. For this, it must specify actions called *obligations* ensuring the security of the data.

A purpose involves two kinds of actions:

- *Rights*: a right is an action the provider is allowed to perform. For each right, we specify the *entities* authorized to perform it, the *delay* in which the entities own the right and the *delay* in which the right must be achieved once activated. Also a right can launch a set of *obligations*.
- *Obligations*: an obligation is the action the provider must achieve after the collection of the private data to ensure their security. An obligation is specified like a right but it does not involve any subsequent actions (i.e. other rights or obligations).

Definition 1. Let us define the following sets:

U: set of entities, D: set of data, A: set of actions, P: set of purposes, O: set of obligations, R: set of rights, I: set of intervals.

U, D, A are represented as a hierarchy or an ontology used to compare the level of the restriction of two policies.

(i) A privacy policy is a set of terms of data use (tdu) where tdu is an element of $D \times P$.

(ii) A purpose p is defined by the tuple $(a, u, \mu, S^R, S^O) \in A \times U \times I \times 2^R \times 2^O$, where a is the action identifying the purpose, u is the entity performing the purpose, μ is an interval in which u must perform the action a, S^R is a set of rights and S^O is a set of obligations associated with the purpose.

(iii) A right r is defined by the tuple $(a', u', v', \mu', S^{O'}) \in A \times U \times I \times I \times 2^O$, where a' is the action identifying the right, u' is the entity authorized to perform the action a', v' is the delay in which the entity u' is authorized to perform the action a' and μ' is the delay in which the action a' must be performed once activated.

(iv) An obligation o is defined by the tuple $(a'', u'', v'', \mu'') \in A \times U \times I \times I$ such that a'' is the action identifying the obligation, u'' is the entity performing the obligation, v'' is the delay in which the action a'' is valid and, μ'' is the validity time of the action a'' must be performed once activated.

As the provider can disclose the data it collects to a third party, we distinguish between two kinds of entities: (i) *ours* specifies the entities of the service collecting the private data, and (ii) *others* specify a third party entity for which a service can disclose the collected private data.

Example 1. Back to the motivating example, the client must provide her *Credit Card Number* (*CCN*) to *confirm the hotel reservation*. The restrictions on *CCN* are as follows:

The financial service collects the credit card number *CCN* to pay the corresponding hotel reservation (p_1) within 20 min after the collection of the *CCN*. The Bank which is an external entity owns *the right* to verify the *CCN* validity (r_1) within 15 min after its collection. If the right (r_1) is triggered, its related action must be performed within 5 min. The financial service *must destroy* the *CCN* (o_1) within 10 min after the achievement of the purpose. Moreover, the Bank *must encrypt* the *CCN* (o_2) within 60 min following the *CCN* reception. The destruction (o_1) and the coding (o_2) must be performed immediately after their triggering, specified by an empty interval time $[0, 0]$. The corresponding policy for *CCN* is given in Fig. 2.

$plcy = \{(CCN, p_1)\}$ where
$p_1 = (PayReservation, Ours : FinancialService, \mu : [0,20mn], \{r_1\}, \{o_1\})$
$r_1 = (ValidityVerification, Others : Bank, v : [0,15mn], \mu : [0,5mn], \{o_2\})$
$o_1 = (Destruction, Ours : FinancialService, v : [t(p_1), t(p_1) + 10mn], \mu : [0,0mn])$
such that $t(p_1) \in [0,20mn]$ is the time when the purpose was fulfilled.
$o_2 = (Encrypt, Others : Bank, v : [0,60mn], \mu : [0,0mn])$

Fig. 2. CCN Policy.

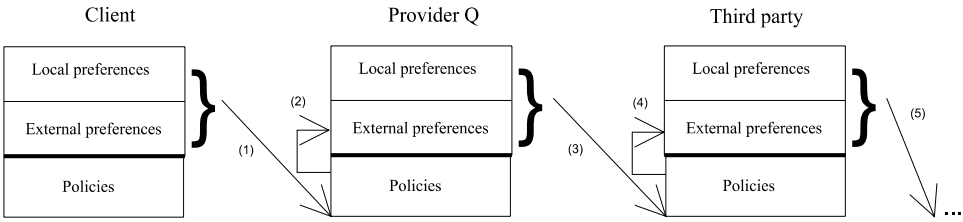


Fig. 3. Extraction of the external preferences from policies.

2.2. *Preferences*

A client service can send its own private data to a provider, so it should specify through rules called *local privacy preferences* how it wishes the provider to use these private data.

Since a service Q can invoke other services, it can disclose the private data of its client to a third party, as depicted in Fig. 3. Thus, it should specify through rules, called *external preferences*, how the third party must use these private data. The *local* and *external preferences* constitute the *preferences* of the client (see (1) in Fig. 3).

The interaction between the service Q and a third party is based on the service preferences of Q and the policies of the third party ((3) in Fig. 3). To inform the third party of the client restrictions through the Web service Q, we add these restrictions in the external preferences of the service Q ((2) in Fig. 3) as depicted in Fig. 3.

Thus, we propagate the set of *tdu* of the service Q policy to external preferences ((4) in Fig. 3). To this aim, we add each *tdu* of the policy to the set *TDU* of preferences when the entity responsible to fulfill a purpose (right or obligation, respectively) is an external entity (*Others*). The propagation keeps continuing for each triggered preferences.

Moreover, in a privacy policy, a preference is defined by a finite set of terms of data use. However, we do not distinguish between the internal (*Ours*) and external (*Others*) entities. In fact, the client ignores whether the entities are considered as a third party (*Others*) for the service collecting the private data or not (*Ours*).

3. Privacy-Aware Business Protocols

Business protocols are becoming a necessary part of Web services description. In fact, the service description not only includes the interface definition and the transport-level properties which can be specified in WSDL. It may also include a business protocol definition that specifies the possible message exchange sequences (conversations) that are supported by a service (see Refs. 7 and 15), or a business process, either from a local or global point of view, respectively referred to as orchestration- and choreography business protocols.[16] The interactions between clients and services are often structured in terms of a set of operation invocations,

whose order typically has to obey certain constraints for clients to be able to obtain the service they need. Business protocols can be specified using notations such as BPELlight,[17] and more generally any formalism that describes the interactions between and among participants made of message exchanges.

Furthermore, developers of client applications need to be aware not only of functional aspects but also of nonfunctional aspects, including privacy. Indeed, the major concerns of a client are the disclosure of its personal data conveyed (if so) during the message exchange. In this section, we first recall the model of business protocols we want to extend to accommodate privacy,[7] and then we describe the integration framework.

3.1. *Business protocols*

A business protocol aims at specifying the external behavior (the sequences of supported messages) of a Web service. In Ref. 7, the authors highlighted the fact that in order to help service development and interoperability, there is a need for formal methods and software tools allowing an automated analysis of service description in order to:

- Identify which conversation can be carried out between two services (compatibility analysis).
- Manage service evolution (replaceability analysis).

The authors have developed a simple and expressive business protocol model based on state machines, an algebra for business protocol analysis, and a set of operators to compare and manipulate protocols. The authors have implemented the framework within ServiceMosaic platform[a] a CASE tool environment that enables the model design, development and management of Web services.

A business protocol in this context is defined in terms of sequences of states and messages. It is specified as a finite state machine, which is built upon traditional state-machine formalism. It is commonly used to model protocols and the external behavior of systems. In the model, states represent the different phases that a service may go through during its interaction with a requester. The states correspond to different states of the service and the transitions correspond to exchanged messages (input and output messages). Each transition is labeled with a message name followed by the message polarity indicating whether the message is incoming (plus [+] sign) or outgoing (minus [−] sign) (see Refs. 8 and 18).

Transitions are triggered when the associated message is sent or received, depending on the polarity. A message corresponds to an invocation of a service operation or its reply. For instance, the protocol depicted in Fig. 4, specifies that the hotel booking service is initially in the Start state, and the client starts to use the service by sending a login message, upon which the service moves to the Logged state transition (login(+)). The figure depicts the sequence of message

[a]http://servicemosaic.fr.

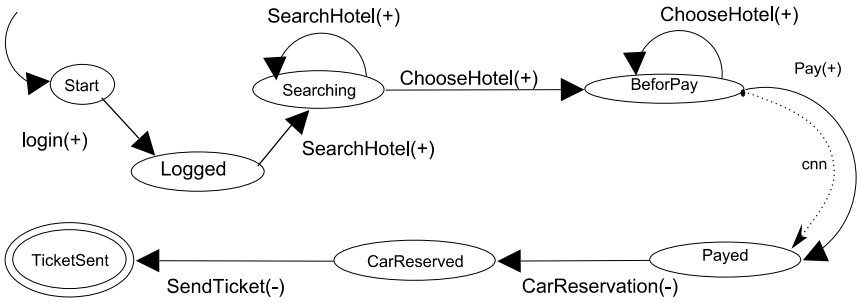


Fig. 4. A hotel booking business protocol.

Login(+).SearchHotel(+).ChooseHotel(+).Pay(+) i.e. a conversation supported by the protocol. As we can see the private data *CCN* is used in the conversation messages.

3.2. *Privacy-aware business protocols*

As shown above, personal data can be handled while exchanging messages between client and provider. In order to better accommodate this fact, we extended the polarity function in a business protocol to consider different privacy aspects and redefined the transitions and states according to this extension. We hence build a new type of business protocol called *privacy-aware business protocol*. For this purpose we annotate and verify the compliance of business protocols with respect to the privacy policies and preferences.

In a privacy-aware business protocol, an interaction between a client and a service provider is established as follows: The client specifies to the service provider its preferences with respect to its private data and how these data can be used. This specification represents the *privacy preferences* of the client. Using its set of *privacy policies*, the service provider attempts to interact with the client while satisfying its privacy preferences. A conversation between a client and a provider can be initiated only if the client's preferences are consistent with the provider's privacy policies. A preference is consistent with a policy if it is less restrictive than the policy. The restriction level of two given policies is described in Ref. 3.

In what follows, we show how to extend business protocols to accommodate privacy rules. More specifically, our first extension consists in extending the polarity function to consider the different aspects of privacy. Moreover, we extend the definition of transitions and states of business protocols.

Polarity. To specify that the input (respectively the output) message imports (respectively exports) private data (for short, we say private message), we propose to extend the polarity function by distinguishing between two variants according to the types of messages:

- Polarity of incoming messages: This polarity indicates if the message imports private data of the clients or not.

- Polarity of outgoing messages: This polarity indicates if the message exports its own private data or those of its clients or not.

Transitions. In business protocols, privacy policies must be associated with input private messages. Therefore, we propose to annotate each transition enabling an input private message by the corresponding policy.

States. In business protocols, a state can be a source for a set of transitions enabling an output private message. Hence, the corresponding preferences are associated with this state.

The following definition represents an extended definition of protocols integrating privacy aspects.

Definition 2. A privacy-aware business protocol Q is a tuple $Q = (S, s_0, F, M, PREF, \vartheta, PLCY, T)$ which consists of the following components:

- S is a finite set of states, where $s_0 \in S$ is the initial state.

$F \subseteq S$ is a set of final states. If $F = \varnothing$, then Q is said to be an empty protocol.

- M is a finite set of messages. For each message $m \in M$, we define two variants of the function $Polarity(Q, m)$:

The polarity of input messages has the form $IPolarity(Q, m) = (+, ClientPData)$, such that:

$$ClientPData = \begin{cases} 1, & \text{if the message } m \text{ imports private data of clients} \\ 0, & \text{otherwise.} \end{cases}$$

The polarity of output messages has the form

$$OPolarity(Q, m) = (-, ClientPData, ServicePData),$$

such that:

$$ServicePData = \begin{cases} Y, & \text{if the message } m \text{ exports private data of the service} \\ N, & \text{otherwise.} \end{cases}$$

and *ClientPData* as above.

- *PREF is a finite set of preferences.*
- $\vartheta : S \rightarrow 2^{PREF}$ *assigns a set of preferences to states.*
- *PLCY is a finite set of policies.*

$T \subseteq S^2 \times M \times (PREF \cup PLCY)$ is a finite set of transitions. Each transition $(s, s', m, plcy)$ identifies a source state s, a target state s', a message m, the corresponding policies $plcy \subseteq PLCY$ (if m is an input private message) and the corresponding preferences $\vartheta(s)$ assigned to the state s (if m is an output private message). In this case, we say that the message m is enabled from a state s:

- *If m is an input private message, then* $plcy \neq \varnothing$.
- *If m is an output private message, then* $\vartheta(s) \neq \varnothing$.

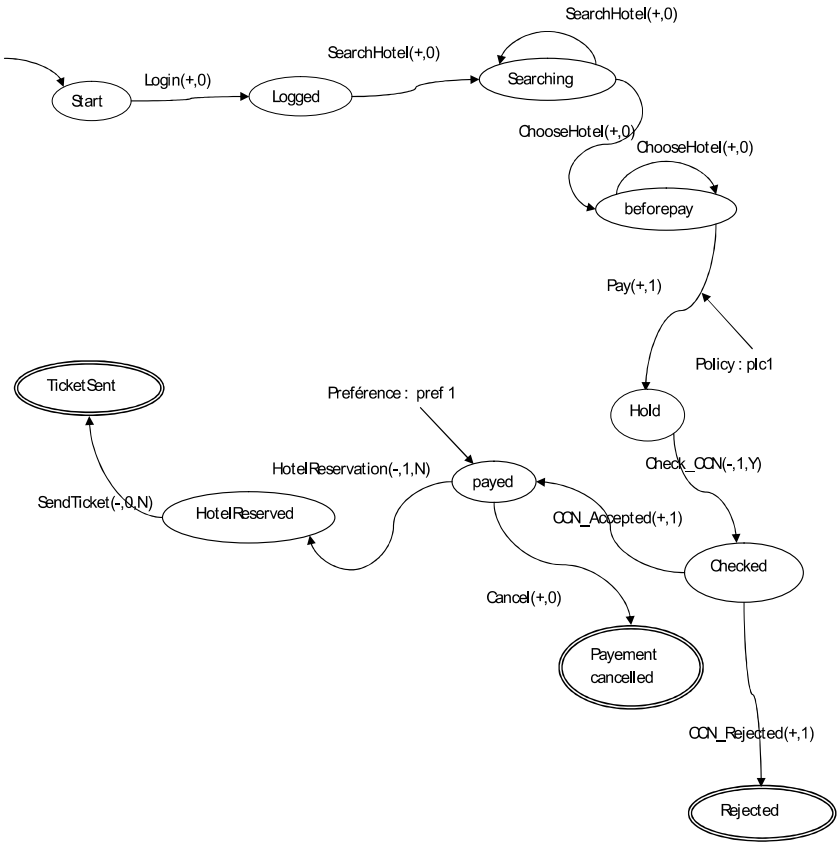


Fig. 5. Privacy-aware business protocol.

Example 2. Figure 5 shows a graphical representation of a privacy-aware business protocol Q of a Web service that allows the reservation of a hotel, presented in the motivating example. The message $Pay(+,1)$ is an input message $(+)$ which imports a private data (the CCN — see (1) in Fig. 5). Hence, we annotate the corresponding transition by the adequate policy. Moreover, the message $CarReservation(-,1,N)$ is an output message $(-)$ which exports the private data of clients (1) and does not export its own private data of the service (N). So, we annotate the source state of this transition by the corresponding preferences as follows: $pref = \{(Name, p'_1)\}$ such that:

$$p'_1 = (CarReservation, u : ReservationAgency, \mu : [0, 20\,\text{min}], \{o'_1\})$$
$$o'_1 = (Destruction, u : ReservationAgency, v : [t(p'_1), t(p'_1) + 10\,\text{min}], \mu : [0, 0\,\text{min}])$$

such that $t(p'_1) \in [0, 20\,\text{min}]$ is the time when the purpose p'_1 was fulfilled.

Consider that a private data d is handled by two input messages m_1 and m_2 of a provider service. We associate to the message m_1 the corresponding policy $plcy_1$

and to the message m_2 the corresponding policy $plcy_2$. Suppose that the preference of the client (the one that invokes this provider service) is consistent with the policy $plcy_1$ and is inconsistent with the policy $plcy_2$. In our approach, this inconsistency does not forbid the conversation between the client and the provider unless the invoked message is m_2. However, the traditional approaches consist in checking all rules (a global policy) related to the data d which prevents the conversation between the client and the provider even if the rule does not deal with the invoked message.

Simulation of privacy-aware protocols

Informally speaking, a privacy-aware business protocol Q is simulated (replaced) by a protocol Q' if, starting from the initial state, each input (respectively, output) message of Q can be matched with an input (respectively, output) message of Q'.[6] Here, we introduce an extended definition of simulation we have proposed, to cater with privacy policies.

Definition 3. Let $Q = (S, s_0, F, M, PREF, \vartheta, PLCY, T)$ and $Q' = (S', s'_0, F', M', PREF', \vartheta', PLCY', T')$ be two protocols. The protocol Q' simulates the protocol Q denoted $Q \prec Q'$ if there exists a relation $\Gamma \subseteq S \times S'$ such that the following holds:

- $\forall (s_1, s'_1) \in \Gamma$ and $\forall T(s_1, s_2, m, plcy), \exists\, s'_2$ such that $T'(s'_1, s'_2, m', plcy'), m \subseteq m', Polarity(Q, m) = Polarity(Q', m'), (s_2, s'_2) \in \Gamma, plcy \prec plcy'$.
- $\forall (s, s') \in \Gamma$, if $s \in F$ then $s' \in F'$.

Remark 1. In order to replace a Web service Q, the first step consists in discovering a set of candidate Web services. Suppose that among all of the discovered Web services, there is no service fully replacing Q according to its private policies. It can thus be very interesting to choose from all of these services the one having the best neighboring policies (that is, ensuring the highest level of restrictions with respect to the policies of Q). As the neighboring replaceability can be partial, it will not be automatic and will require the authorization of the client. The full study of neighboring replaceability goes beyond the scope of this paper.

When we replace a Web service Q by a new one Q', Q' will use its own private data, so we do not have to check the local preferences. But Q' may also use the private data of the clients of Q, so obviously we have to check the consistency between the external preferences of Q and Q'. Since the external preferences are extracted from policies already checked in the private replaceability of Q by Q', they are also implicitly checked.

Moreover, to better deal with the verification of the compliance of the privacy policies and preferences in such protocols, the adopted model should enable the expression of temporal constraints. Indeed, the management of temporal constraints with respect to time intervals of purposes (respectively rights and obligations) is an interesting opportunity to increase the efficiency of the model. Therefore, we

propose to extend the previous model by considering time-related properties for privacy policies in business protocols.

4. Timed Privacy-Aware Business Protocols

This section introduces the model of *TPBP* which extends privacy-aware business protocols with time-related aspects. Since our privacy-aware business protocol was built upon the traditional state-machine formalism, existing results in timed automata theory can be reused and/or extended to deal with our specific problems. This choice was also motivated by the use of this theory in business protocols.[18] Briefly, a timed automaton[19] is a finite automaton augmented with a finite set of real-valued clocks. Clock constraints can be associated with transitions and can also be reset to zero simultaneously with any transition.

Clock constraints. We equip the *TPBP* with a set of n clock variables, where n is the number of the transitions such as each transition has a clock. Each clock is a variable taking its values from time domain $\mathbf{T}$. The values of these variables change simultaneously. The clock variables do not have the same values at a given time.

For a set of clocks C, the set of clock constraints $\Phi(C)$ is defined by:

$$\varphi_1, \varphi_2 := h \propto k | \neg\varphi_1 | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \mathit{True}$$

with $h \in C, k > 0$ and $\propto \in \{<, \leq, >, \geq, =\}$. For instance, some clock constraints on $C = \{c_1, c_2\}$ are $c_1 = 4, (c_1 < 10) \wedge (c_2 = 5)$.

Reset timers to zero C_0. Each transition in the *TPBP* has a clock. This clock must be incremented while the transition is taking place for the first time. It is reset to zero each time the transition is fired again.

$C \supset C_0 = \{C_i \,|\, \forall\, 1 \leq i \leq n, n_{tr_i} > 1\}$, where C_i is the clock variable associated to the transition tr_i, and n_{tr_i} is the number of triggering transition tr_i.

Clock valuations. A clock valuation on C is an application $v : C \rightarrow \mathbf{T}$. $\mathbf{T}^C$ denotes the set of all clock valuations. The valuations are defined as *passing time* and *reset to zero* operations:

- Let $t \in \mathbf{T}$; we denote $v + t$ the valuation such as

$$(v + t)(c) = v(c) + t, \quad \forall\, c \in C.$$

- Let $r \subseteq C$; we denote $v[r \leftarrow 0]$ the valuation such as:

$$v[r \leftarrow 0](c) = \begin{cases} 0, & \text{if } c \in C_0 \\ v(c), & \text{else} \end{cases}, \quad \forall\, c \in C.$$

The following definition extends the definition of privacy-aware business protocols by incorporating time-related aspects.

Definition 4. Timed Privacy-aware Business Protocol (TPBP)

A *TPBP* is a 10-tuplet $(S, s_0, F, M, T, \vartheta, C, C_0, PLCY, PREF)$ which consists of the following elements:

- S is a finite set of states, where $s_0 \in S$ is the initial state.
- $F \subseteq S$ is a set of final states.
- M is a finite set of messages. For each message $m \in M$, we define two kinds of polarity as defined previously.
- $T \subseteq S \times \varphi(C) \times M \times S$ is a finite set of transitions; $tr = (s(pref), \varphi, m, plcy, s') \in \mathbf{T}$ represents a transition from s to s', $m \in M$ is a message, φ is a clock constraint associated with the transition tr, the policy $plcy \subseteq PLCY$ and the preference $pref \subseteq PREF$. This transition is represented by:

$$s(pref) \xrightarrow{m,\varphi,plcy} s'.$$

- $\vartheta : S \to 2^{PREF}$ assigns a set of preferences to states.
- C a finite set of clocks (with positive real values).
- $C_0 \subseteq C$ is the subset of reset to zero timers.
- PLCY is a finite set of policies.
- PREF is a finite set of preferences.

Depending on the exchanged message m, we distinguish between several transition types:

- If the message m is an incoming message, and its parameters contain private data of the clients, then we have $plcy \neq \varnothing, pref = \varnothing$ and the transition is defined as: $s \xrightarrow{m,\varphi,plcy} s'$.
- If the message m is an incoming message and its parameters do not contain private data of the clients, then we have $plcy = \varnothing, pref = \varnothing$ and the transition is determined by: $s \xrightarrow{m,\varphi} s'$.
- If the message m is an outgoing message, and its parameters contain private data of the clients and/or private data of the service, then we have $pref \neq \varnothing, plcy = \varnothing$ and the transition is defined by: $s(pref) \xrightarrow{m,\varphi} s'$.
- If the message m is an outgoing message, and its parameters do not contain private data of the clients and/or the private data of the services, then we have $pref = \varnothing, plcy = \varnothing$ and the transition is defined as: $s \xrightarrow{m,\varphi} s'$.

***TPBP* semantics.** A state or a configuration of a timed privacy-aware business protocol is defined as $(s, v) \in S \times \mathbf{R}_+^C$, where s and v represent the current state and the valuation of the clocks respectively. $\mathbf{R}_+$ is a set of non-negative real numbers.

A pair (s, v) is considered as an initial state of the *TPBP* if $s = s_0$ and $v(C) = 0$ for each clock in C. The execution of a *TPBP* is a sequence of (message, time) pairs:

$$\sigma = (s_0, v_0) \xrightarrow{m_1,t_1} (s_1, v_1) \xrightarrow{m_2,t_2} (s_2, v_2) \xrightarrow{m_3,t_3} \cdots (s_n, v_n)$$

with $t_i, t_{i+1} \in \mathbf{R}_+$ and $t_{i+1} \geq t_i$ for each $i > 0$. Time t_i corresponds to the passing time of m_i, and $v_i \geq 0$ represents the clock valuations such that:

- $v_0(c) = 0, \forall\, c \in C$.
- $\forall\, i > 0, v_{i-1} + (t_i - t_{i+1}) \models \varphi_i$.
- $\forall\, i > 0$ and $\forall\, c \in C, v_i = \begin{cases} 0, & \text{if } c \in C_0 \\ v_{i-1} + (t_i - t_{i+1}), & \text{otherwise} \end{cases}$

Given the above formal model of *TPBP*, in the next section we will discuss how to check the compliance of policies while replacing service providers.

5. TPBP Verification

To ensure a better analysis of a *TPBP* and check the satisfaction of its time-related properties, we provide in this section a categorization of the *TPBP* and an identification of the properties to be checked.

5.1. *Categorization of privacy-aware business protocols*

We distinguish between two *TPBP* types: current *TPBP* and transversal *TPBP*.

Definition 5. Current *TPBP*
A current TPBP, denoted by $TPBP_C$, collects the private data quoted in the policy and used for the current operation in the state.

For instance, the current business protocol of Fig. 6 collects the email addresses in order to send the ticket to the client.

Definition 6. Transversal *TPBP*
A transversal TPBP, denoted by $TPBP_T$, uses the private data collected by the $TPBP_C$.

For instance, the email addresses collected by the $TPBP_C$ can be used by another business protocol to send information related to discounts. This latter protocol acts as a transversal protocol. Protocols $TPBP_2$ and $TPBP_3$, depicted in Fig. 6, represent two transversal protocols detailed in Fig. 7. They use two private data collected by $TPBP_1$ (email and CCN, respectively).

The relationship between the classes of protocols is defined as follows:

Definition 7. Transition link
Let $TPBP_C$ be a current protocol and $TPBP_T$ its corresponding transversal protocol. The link between them is defined as follows:

$$TL_{ij} : TPBP_C \rightarrow TPBP_T,$$
$$s_i \mapsto q_{r_j},$$

where s_i is a state in $TPBP_C$ and q_{r_j} is a state for which the right r_j holds in $TPBP_T$. We denote by $L_i = \{TL_{ij}, \forall\, 1 \leq j \leq j\}$ the set of transition links between

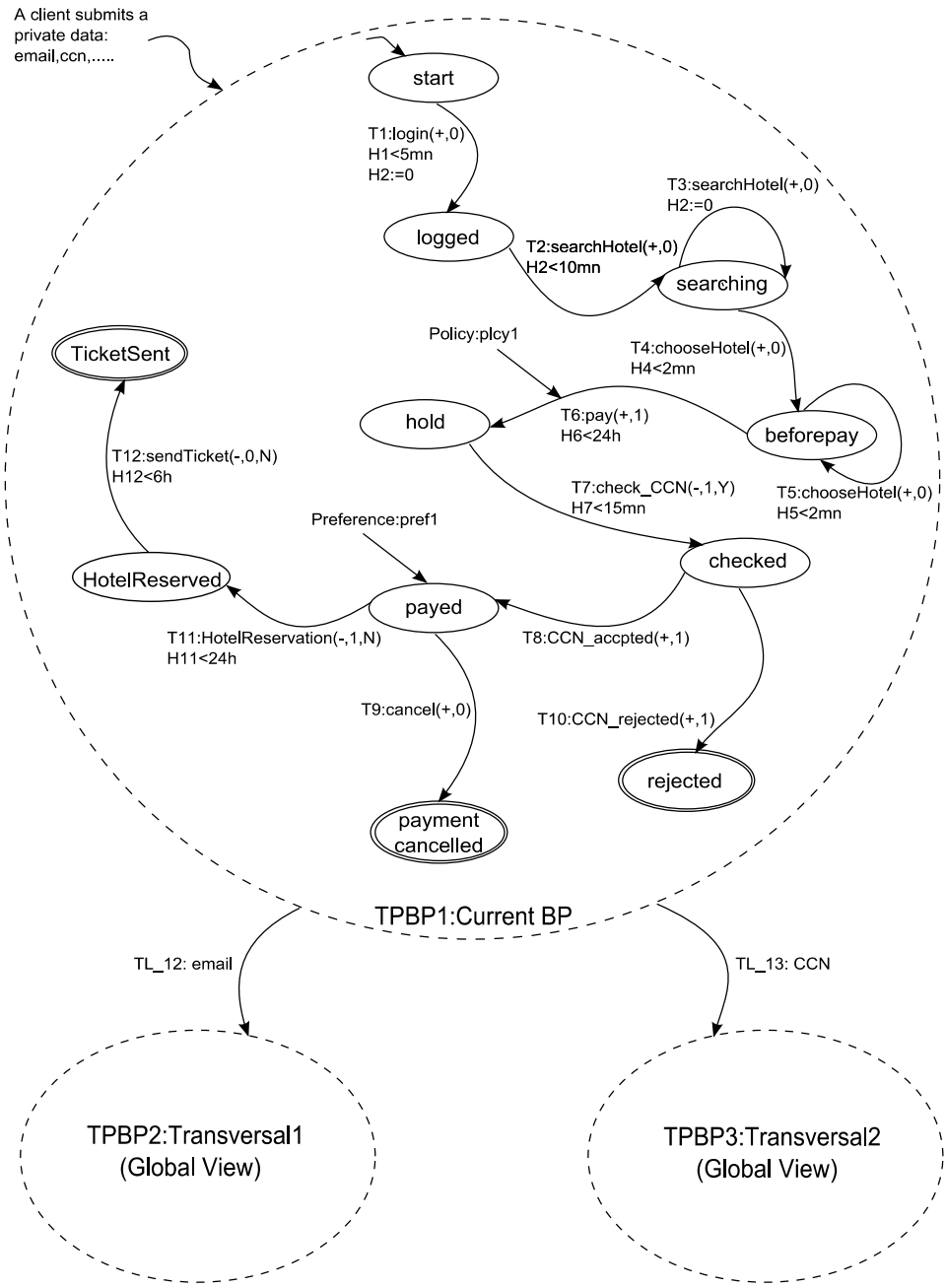
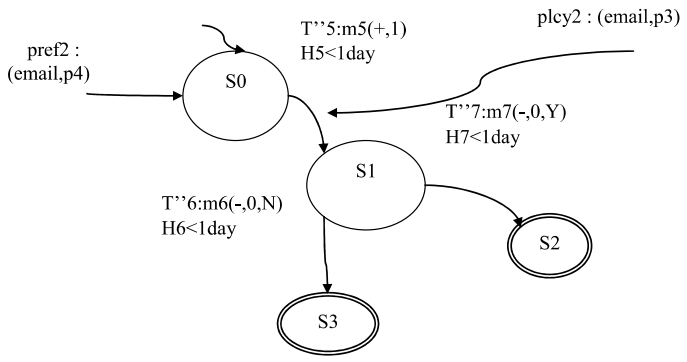


Fig. 6. Types of timed privacy-aware business protocol.



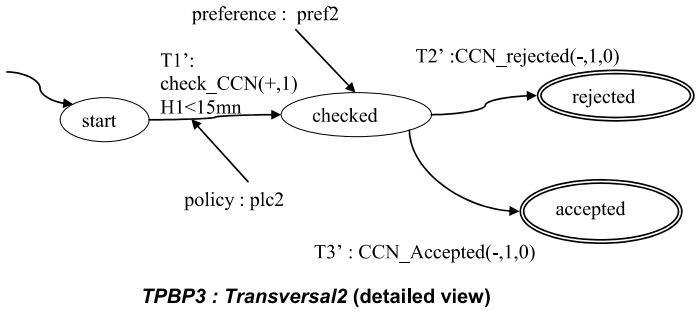


Fig. 7. Two transversal privacy-aware business protocols.

the state s_i of $TPBP_C$ collecting a private data d and the states q_{r_j} of $TPBP_T$ using d.

5.2. *TPBP constraints*

In what follows, we define two kinds of constraints to be verified: clock constraints and privacy constraints.

Clock Constraint. The clock constraint can be interpreted naturally through the valuation: If v is a valuation and $(v(c))_{c\in C}$ satisfies the clock constraint φ we say $(TPBP, s, v)$ verifies φ, and we write $(TPBP, s, v) \models \varphi$, such that:

- $(TPBP, s, v) \models c \propto K$ iff $v(c) \propto K$.
- $(TPBP, s, v) \models \neg\varphi$ iff $(S, v)| \neq \varphi$.
- $(TPBP, s, v) \models \varphi_1 \wedge \varphi_2$ iff $(TPBP, s, v) \models \varphi_1$ and $(s, v) \models \varphi_2$.
- $(TPBP, s, v) \models \varphi_1 \vee \varphi_2$ iff $(TPBP, s, v) \models \varphi_1$ where $(TPBP, s, v) \models \varphi_2$.

Privacy constraints. Let $T(s_{i-1}, m, s_i), \forall\, i > 0$, be a transition in $TPBP$, where m is an incoming message containing private data. Therefore, a policy *plcy* is associated to the transition. We define δ_i as a clock associated with the policy, leading

to state s_i and initialized to zero. It can be reset to zero after firing the transition T again. The valuation of the policy clock is defined by $v(\delta_i)$.

Purpose. Let $\{op_1, \ldots, op_k\}$ be the set of operations related to the purpose of current state s_i. Let $f : S \rightarrow 2^{purpose}$ be a labeling function associating to each state s_i, $f(s)$ the set of purpose operations satisfied in the state $s_i, \forall\, s_i \in S$. Then:

$$(TPBP, s_i) \models op_1 \wedge op_2 \wedge \cdots op_k \quad \text{iff } op_j \in f(s_i),$$

and $Inf(bound) \leq v(\delta_i) \leq Sup(bound)$, where $Inf(bound)$ is the lower bound of time interval of the operation validity, $Sup(bound)$ is the upper bound of time interval, and $v(\delta_i)$ is the valuation of policy clock at the execution time point of operation op_j.

Obligation. Let $O_i = \{o_1, \ldots, o_l\}$ be the set of obligations associated to purpose p_k in state s_i. $OP_{O_i} = \{op(o_1), \ldots, op(o_l)\}$ is the set of operations related to obligation O_i. Let $\eta : S \rightarrow 2^{obligation}$ be a labeling function associated with each state s_i, where $\eta(s)$ is the set of operations related to obligations checked in s_i. Then, the following holds:

$$(TPBP, s_i) \models op(o_1) \wedge op(o_2) \wedge \cdots op(o_l) \quad \text{iff } op(o_j) \in \eta(s_i),$$

and $Inf(\lambda_i) \leq v(\delta_i) \leq Sup(\lambda_i)$ where $Inf(\lambda_i)$ and $Sup(\lambda_i)$ are the lower and upper bounds of λ_i, respectively; and $v(\delta_i)$ is the valuation of policy clock at the execution time point of operation op_j.

Right. Let $OP_{R_i} = \{op(r_1), \ldots, op(r_k)\}$ be the set of operations associated to rights. Let $g : S_T \rightarrow 2^{right}$ be a labeling function associating to each state $s_i \in S_T$, $g(s)$ an operation associated to the right and satisfied in s_i. Then:

$$(TPBP_C, s_i) \models op(r_1) \wedge op(r_2) \wedge \cdots op(r_k) \quad \text{iff } TPBP_{T(r_i)}, q(r_i) \models op(r_i),$$

with $TPBP_{T(r_i)}, q(r_i) \models op(r_i)$ iff $op(r_i) \in g(q(r_i))$ and $Inf(\lambda) \leq v(\delta_i) \leq Sup(\lambda)$, where $op(r_i)$ is an operation associated to the right r_i, $Inf(\lambda)$ and $Sup(\lambda)$ are lower and upper bounds of λ, respectively; and $v(\delta_i)$ is the valuation of the policy clock at the execution time point of operation $op(r_i)$.

Lemma 1. *Let s_i be a state in $TPBP_C$, $q(r_j)$ a state in $TPBP_{T(r_j)}$ for which the right r_j is associated and $op(r_j)$ the operation associated with the right r_j then*:

$$(TPBP_C, s_i) \models op(r_j) \quad \textit{iff } TPBP_{T(r_j)}, q(r_j) \models op(r_j),$$

where $q(r_j) = TL_{ij}(s_i)$.

5.3. *Verification steps*

Before defining the verification steps, let us first introduce the *TPBP* path notion:

Definition 8. *TPBP* paths
A path p in *TPBP* is defined in terms of states and transitions as follows:

$$p = s_0 \xrightarrow{\varphi_1, m_1} s_1 \xrightarrow{\varphi_2, m_2} s_2 \xrightarrow{\varphi_3, m_3} \cdots s_n,$$

where s_0 is the initial state, s_n is the final state, and $\forall 0 < i \leq n, (s_{i-1}, \varphi_i, m_i, s_i) \in \mathbf{T}$.

Definition 9. An execution x in *TPBP* on the path p is defined as:

$$x = (s_0, v_0) \xrightarrow{\varphi_1, m_1, t_1} (s_1, v_1) \xrightarrow{\varphi_2, m_2, t_2} (s_2, v_2) \xrightarrow{\varphi_3, m_3, t_3} \cdots (s_n, v_n),$$

with $\forall 0 < i \leq n, v_i \in \mathbf{T}^C$ and $(t_i)_{i>0}$ a temporal sequence. We denote by $x(s_0)$ the set of executions started from the initial state s_0.

The constraint verification consists of the following phases:

Phase 1. This phase deals with the verification of the states along the execution path. If m_i is an incoming message to s_i containing private data, the following elements need to be checked:

(i) the purpose associated to the state s_i,
(v) the obligations $(o_1, o_2, \ldots, o_l)$ associated with p,
(vi) the rights $(r_1, r_2, \ldots, r_k)$ associated with p, and
(vii) the obligations $(o_{j1}, o_{j2}, \ldots, o_{jq})$ associated with rights r_j, $\forall 1 \leq j \leq k$.

The behavior of a state s_i is satisfied iff

$$\begin{aligned}((TPBP_C, s_i) \models op(p) \wedge (TPBP_C, s_i) \models op(o_1) \wedge op(o_2) \wedge \cdots op(o_l)) \\ \wedge (\forall 1 \leq j \leq k, (TPBP_C, s_i) \models op(r_j) \wedge (TPBP_C, s_i) \models op(o_{j1}) \\ \wedge op(o_{j2}) \wedge \cdots op(o_{jq})).\end{aligned}$$

Phase 2. This phase deals with the verification of the clock constraints associated with the transitions along the execution paths. A clock constraint is satisfied iff $(TPBP_C, s_i, v_i) \models \varphi_i$, where φ_i is the clock constraint associated with the transition and s_i is the destination state of the transition.

Phase 3. This phase considers the conversation in the current business protocol. More precisely, given the initial and final states s_i and s_n in the execution path:

$$r = (s_0, v_0) \xrightarrow{\varphi_1, m_1, t_1} (s_1, v_1) \xrightarrow{\varphi_2, m_2, t_2} (s_2, v_2) \xrightarrow{\varphi_3, m_3, t_3} \cdots (s_n, v_n),$$

the conversation is satisfied iff s_0 is the initial state, $s_n \in F$ and $\forall 1 \leq i \leq n$, $(TPBP_C, s_i, v_i) \models \varphi_i \wedge (TPBP_C, s_i) \models op(r_i)$.

To conclude, while web services are exchanging messages, some personal data are likely involved in the conversation of different kinds of business protocols. Hence, it was necessary to integrate a privacy model in business protocol, in order to check the validity of personal data usage. Moreover, we emphasize that time is a crucial abstraction that has not been studied up to date in the area of privacy. That is, given the importance of considering time-related properties, we presented concepts and techniques, for performing time-related analysis and verification of policies in TPBP.

6. Related Work and Discussion

In recent Web services research there is an increasing demand and discussion about privacy technologies for supporting different business applications. For example, WS-Policy describes the business policies to be enforced on intermediaries and endpoints.[12] The current WS-Policy specification does not discuss privacy rules in details. Even though WS-Privacy is proposed as a model for defining subject privacy preferences and organizational privacy practice statements, WS-Privacy has not been fully developed yet.[12] The EPAL technical specification is used to formalize privacy authorization for actual enforcement within an intra or inter-enterprise for business-to-business privacy control.[14] However, it does not consider privacy enforcement in the context of the WSA.

The P3P(1) developed by the W3C enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by agents like Web browsers. APPEL (A P3P Preference Exchange Language)[10] provides a standard way of defining the user privacy preferences in a set of preference rules, which can be used by the user agent to make automated and semi-automated decisions regarding the acceptance of privacy policies from P3P-enabled Web sites.

There are a few number of research works related to Web services privacy policies. In Ref. 20, the authors focus on Web service privacy issues; they do not discuss the related technical issues in the context of WSA. In Ref. 21, the authors present a privacy framework for Web services which allow user agents to automatically negotiate with Web services on the amount of personal information to be disclosed on behalf of the user. Semantic issues for privacy management have also been discussed in some works. Reference 22, points out that a standard method of exchanging privacy policies, that is, a privacy ontology, is needed for the Semantic Web. Reference 23 defines a vocabulary for composing policies to allow or deny access to the personal information that a policy governs.

In Ref. 24, the authors address security of semantic Web services that are declaratively described in OWL-S. They propose ontologies to annotate OWL-S input and output parameters with respect to their security features, including encryption and digital signatures. Moreover, they propose to incorporate privacy and authentication policies into OWL-S descriptions and requester profiles. They designed and implemented algorithms to check policy compliance and integrated them in the service selection process of the OWL-S MatchMaker. This is done by extending the OWLS VM with features for encrypting and signing messages exchanged between service requester and provider. In Ref. 25, the authors present the KAoS toolset for the specification, management, analysis, disclosure and enforcement of policies represented in OWL. They discuss three current Semantic Web Service applications as examples of the kinds of roles that a policy management framework can play: as an authorization service in grid computing environments, as a distributed policy specification and enforcement capability for a semantic matchmaker, and as a verification tool for service composition and contract management.

In Ref. 26, the authors propose a Web service privacy framework based on a policy approach enhanced with ontologies. It uses different Web standards for supporting privacy protection, including P3P, the Web services policy framework (WS-Policy) and the Web ontology language (OWL). In Ref. 27, the authors define a meta-model for privacy policy creation and comparison for Web services based on fair information practices introduced around the world to protect the privacy of individuals. They developed criteria for the comparison of the elements that compose the policies, creating hierarchical relationships between those elements that could not otherwise be directly compared.

In Ref. 28, the authors propose a framework to support user control over the data made available to service providers in the context of an OSGi (Open Services Gateway initiative)-based Extensible Service Systems. A formal privacy model is defined and service and policy descriptions are deduced. Technical system requirements to support those policies are identified. Since guaranteeing privacy inside the system is of little help if any malicious entity can break in to it, security architecture for OSGi-based Extensible Service Systems is also defined. The framework is a compound of two elements: an architecture for secure interactions between the users and the pervasive system, and a meta-data language that expresses privacy properties of services and user-defined policies. All these works are concerned with data semantics but do not deal with service behavior as we are doing in our work.

On the other hand, several ongoing efforts in the area of Web services recognize the importance of high level modeling and analysis of service protocols. In Ref. 18, the authors proposed to study the problem of automated analysis of Web services protocol compatibility and replaceability in presence of timing abstractions based on timed automata. The model supports rich timing constraints but only deals with functional aspects of the service. Reference 9 has addressed the problem of functional qualitative and quantitative analysis of timing aspects of Web service compositions defined as a set of BPEL4WS. The authors introduced a formalism, called Web Services Timed state Transition Systems (WSTTS), to capture the timed behavior of composite Web services. The work presented in Ref. 29 is related to the run-time monitoring of Web service compositions, where the authors define a language for the specification of instance and class monitors. The language allows the specification of boolean, statistic, and time-related properties. The reader may refer to Ref. 18 for more information about business protocols.

In Ref. 30, the author presents an approach for measuring how well a Web service protects personal privacy. To achieve this, the author defined a measure of protection of user privacy as a numerical value that indicates the degree of the user's control (or some aspect of that control) over the service collection, retention and distribution of information about the user. In Ref. 31, the authors propose a framework that addresses consumer privacy concerns in the context of highly customizable composite Web services. Their approach involves service producers exchanging their "terms-of-use" with consumers in the form of "models". The framework provides automated techniques for checking these models at the consumer side for

compliance of consumer privacy policies. In the event of a policy violation, the framework supports automatic generation of "obligations" that the consumer generates for the composite service. These obligations are automatically enforced through a dynamic program analysis approach on the web service composition code.

Moreover, other areas have studied privacy; for instance, in the databases area (see Refs. 32 and 33) discussed issues related to privacy preserving data publishing. In the social networking area, Ref. 34 presented methods to anonymize a dynamic network such that the privacy of users is preserved when new nodes and edges are added to the published network.

In our work, we have considered the two previous aspects (time and privacy) in the same model for business protocol, which allows benefiting from the important results of the proposed approaches and therefore increases the efficiency of our model.

7. Conclusion and Future Work

In the past few years, Web services privacy issues have been attracting more and more attention from the industry and research community. Since the number of Web services-based business applications is increasing, one can imagine that the demands for privacy enhancing technologies for Web services will also be increasing in the future. It is known that a service description should not only include the service interfaces as in conventional middleware, but also the business protocol i.e. the specification of which message exchange sequences are supported by the service. To automate the analysis of service descriptions a simple and expressive business protocol model based on state machines is proposed in the literature, which supports rich timing constraints handling the service functionalities. In this paper, we investigated an extension of business protocols in order to accommodate privacy aspects and time-related properties, leading to what we call *TPBP*s. The proposed model was described as a state machine, emphasizing the privacy requirements and in particular the time-related properties. We also discussed the properties to be checked and described the verification process. We currently are working on the following issues:

- Definition of fine-grained timed properties. For instance, we would like to investigate the combination of time related properties of behaviors with those of privacy.
- What Web services need to know is not only user preferences but also the user context, which includes any information that can be used to characterize the user and her situation. Hence, user context should include user's local data as well as any data stored about the user such as those stored in customer relationship management (CRM) systems to make effective use of Web services.

Acknowledgments

We acknowledge the reviewers for their great effort and consideration in improving the quality of this paper. With their kind assistance, we were able to correct many

inconsistencies and pitfalls. We would also like to thank Mr. Omar ASLAOUI for his suggestions and advises. We also express our gratitude for the editor's expertise and time in polishing (and publishing) this manuscript. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007–2013 under grant agreement 215483 (S-Cube).

References

1. R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, Implementing P3P using database technology, in *Proc. IEEE Int. Conference on Data Engineering (ICDE'03)* (2003), pp. 595–606.
2. M. C. Mont and F. Beato, On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises, in *Proc. IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY 07)* (2007), pp. 51–55.
3. N. Guermouche, S. Benbernou, E. Coquery and M. S. Hacid, Privacy-aware web service protocol replaceability, in *Proc. IEEE Int. Conference on Web Services (ICWS 07)* (2007), pp. 1048–1055.
4. S. Benbernou, H. Meziane, Y. H. Li and M. S. Hacid, A privacy agreement model for web services, in *Proc. IEEE Int. Conference on Service Computing (SCC 07)* (2007), pp. 196–203.
5. T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic and S. Weerawarana, Business Process Execution Language for Web Services (version 1.1) Specification, BEA Systems, IBM Corp., Microsoft Corp. SAP AG, Siebel Systems (2003).
6. B. Benatallah, F. Casati and F. Toumani, Analysis and management of web service protocols, in *Proc. International Conference on Conceptual Modeling (ER 04)* (2004), pp. 524–541.
7. B. Benatallah, F. Casati and F. Toumani, Representing, analysing and managing web service protocols, *Data & Knowledge Engineering Journal* **58**(3) (2006) 327–357.
8. J. Ponge, B. Benatallah, F. Casati and F. Toumani, Fine-grained compatibility and replaceability analysis of timed web service protocols, in *Proc. 26th International Conference on Conceptual Modeling (ER 07)* (2007), pp. 599–614.
9. R. Kazhamiakin, P. Pandya and M. Pistore, Representation, verification, and computation of timed properties in web, in *Proc. IEEE Int. Conference on Web Services (ICWS'06)* (2006), pp. 497–504.
10. L. Cranor, M. Langheinrich and M. Marchiori, A P3P Preference Exchange Language 1.0 (APPEL1.0), W3C Working Draft, no. 15 (2002).
11. P. C. K. Hung, E. Ferrari and B. Carminati, Towards standardized web services privacy technologies, *IEEE Int. Conference on Web Services (ICWS'04)* (2004), p. 174.
12. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker and M. Hondo, Web services policy framework (WS-Policy), W3C recommendation version 1(2), 2003–2006 (2006).
13. L. Cranor, G. Hogben, M. Langheinrich, M. Marchiori, M. Presler Marshall, J. Reagle and M. Schunter, The Platform for Privacy Preference 1.1 (P3P 1.1) Specification, *Technical report* (2005).
14. I. B. M. Corporation, Enterprise Privacy Authorization Language (EPAL), in *IBM Research Report* (2004).

15. B. Benattallah, F. Casati and F. Toumani, Analysis and management of web services protocols, in *Proc. Int. Conference on Conceptual Modeling (ER'04)* (2004), pp. 524–541.
16. M. Mancioppi, M. Carro, W. J. Van denHeuvel and M. P. Papazoglou, Sound multi-party business protocols for service networks, in *Proc. Int. Conference on Service Oriented Computing (ICSOC'08)* (2008), pp. 302–316.
17. J. Nitzsche, T. Van Lessen and F. Leymann, Extending BPELlight for expressing multi-partner message exchange patterns, in *Proc. Int. IEEE Enterprise Distributed Object Computing Conference (EDOC 08)* (2008), pp. 245–254.
18. J. Ponge, B. Benatallah, F. Casati and F. Toumani, Analysis and applications of timed service protocols, *ACM Trans. Softw. Eng. Methodol. Journal (TOSEM)* **19**(4) (2010) 11–49.
19. R. Alur and D. L. Dill, A theory of timed automata, *Journal of Theoretical Computer Science* **2**(126) (1994) 183–235.
20. A. Rezgui, M. Ouzzani, A. Bouguettaya and B. Medjahed, Preserving privacy in web services, in *Proc. Int. Workshop on Web Information and Data Management (WIDM 02)* (2002), pp. 56–62.
21. A. Tumer, A. Dogac and I. H. Toroslu, A semantic-based user privacy protection framework for web services, in *Proc. Workshop on Intelligent Techniques for Web Personalization and Recommendation (ITWP 03)* (2003), pp. 289–305.
22. A. Kim, L. J. Hoffman and C. D. Martin, Building privacy into the semantic web: An ontology needed now, in *Proc. Semantic Web Workshop* (SemHE02) (2002), pp. 3375–3386.
23. R. Lee, Personal Data Protection in the Semantic Web, *ME Thesis MIT USA* (2002).
24. L. Kagal, L. T. Finin and A. Joshi, A policy based approach to security on the semantic web, in *Proc. Int. Workshop on Policies for Distributed Systems and Networks (POLICY 03)* (2003), pp. 402–418.
25. M. Uszok, J. M. Bradshaw, R. Jeffers, M. Johnson, A. Tate, J. Dalton and S. Aitken, Policy and contract management for semantic web services, *In Semantic Web Services: Papers from the 2004 Spring Symposium*, ed. Terry Payne, 24–31. Technical Report SS-04-06. Association for the Advancement of Artificial Intelligence (2004).
26. D. Z. G. Garcia and M. Toledo, A web service privacy framework based on a policy approach enhanced with ontologies, in *Proc. IEEE Int. Conference on Computational Science and Engineering Workshops (ICSE'08)* (2008), pp. 209–214.
27. D. S. Allison, H. F. E. Yamany and M. Capretz, Metamodel for privacy policies within SOA, *Software Engineering for Secure Systems (ICSE'09)* (2009), pp. 40–46.
28. P. Parrend, S. Frénot and S. Hoehn, Privacy-aware service integration, in *Proc. IEEE Int. Workshop on Services Integration in Pervasive Environments (SIPE'07)* (2007), pp. 397–402.
29. F. Barbon, P. Traverso, M. Pistore and M. Trainotti, Run-time monitoring of instances and classes of web service compositions, in *Proc. Int. Conference on Web Services (ICW'06)* (2006), pp. 63–71.
30. G. Yee, Measuring privacy protection in web services, in *Proc. Int. Conference on Web Services (ICWS'06)* (2006), pp. 647–654.
31. W. Xu, V. N. Venkatakrishnan, R. Sekar and I. V. Ramakrishnan, A framework for building privacy-conscious composite web services, in *Proc. IEEE Int. Conference on Web Services (ICWS'06)* (2006), pp. 655–662.
32. X. Xiao, G. Wang and J. Gehrke, Interactive anonymization of sensitive data, *ACM Special Interest Group on Management of Data (SIGMOD'09)* (2009), pp. 1051–1054.

33. X. Xiao, G. Wang and J. Gehrke, Differential privacy via wavelet transforms, in *Proc. IEEE Int. Conference on Data Engineering (ICDE'10)* (2010), pp. 225–236.
34. D. S. SmiritiBhagat, G. Gormode and B. Krishnamurthy, Privacy in dynamic social networks, in *Proc. Int. Conference on World Wide Web (WWW'10)* (2010), pp. 1059–1060.