

Supporting Secure Business Process Design via Security Process Patterns

Nikolaos Argyropoulos^(✉), Haralambos Mouratidis, and Andrew Fish

School of Computing, Engineering and Mathematics,
University of Brighton, Brighton, UK
{n.argyropoulos,h.mouratidis,andrew.fish}@brighton.ac.uk

Abstract. Security is an important non-functional characteristic of the business processes used by organisations for the coordination of their activities. Nevertheless, the implementation of security at the operational level can be challenging due to the limited security expertise of process designers and the delayed consideration of security during process development. To overcome such issues, expert knowledge and proven security solutions can be captured in the form of process patterns, which can easily be reused and integrated to business processes with minimal security-related knowledge required. In this work we introduce process-level security patterns, each of which contains the main activities required for the operationalisation of different security requirements. The introduced patterns are then used as a component of an existing framework for the creation of secure business process designs, the application of which, is illustrated through a working example. A preliminary evaluation of the proposed patterns is conducted via a workshop session.

Keywords: Security requirements · Business process modelling · Security process patterns · Business process security

1 Introduction

Business processes are essential instruments utilised by organisations for the coordination of their activities in order to produce value in the form of products and services [21]. During the design of business processes, in addition to their functional characteristics, a number of non-functional aspects also need to be taken into consideration. Security is one of the most important of such non-functional aspects due to the potential impact of its shortcomings for organisations in terms of finances, reputation and legal compliance [17]. Since the consideration of security during the early design stages of systems is considered highly beneficial [12], specialised security-oriented extensions have been developed for the majority of the established process modelling languages. Nevertheless, capturing the context and rationale behind general and security-related design choices made during process design, is outside of the scope of process modelling languages [5].

Aligning system requirements, as captured by goal models at the organisational level, with process activities at the operational level, augments the traceability between system models of different abstraction levels [6]. Additionally, it helps provide justification for design choices and leads to more robust and context-aware operationalisations of security [19]. Therefore, to enhance the alignment between an organisation's strategic goals and its operations, there needs to be a well-defined interconnection between a system's requirements and its process models.

Another obstacle in the design of secure business processes is the disconnect between security experts and the system developers [10]. Since the main concern of system developers is functionality, security is underprioritised and implemented in an ad-hoc manner during the later development stages. Security patterns are often utilised as a way to overcome such issues, as they are able to provide to non-experts standardised and proven solutions to common security-related issues [7]. Patterns can encapsulate security expertise and standardise proven solutions to recurring problems [10], which can facilitate a systematic and structured approach towards the operationalisation of security by non-experts [16].

In this work we introduce a number of reusable process fragments which can be integrated into business process models in order to operationalise different types of security requirements (e.g., confidentiality, integrity, availability). Each of the proposed fragments forms a business process level pattern, generic enough to be able to be instantiated by different types of security implementing technologies (e.g., the authentication pattern can be instantiated by user credentials, biometrics or smart card technologies). The introduced patterns are utilised as a component of an existing framework for the design of secure business processes. Moreover, an initial evaluation of their perceived usability and comparison to ad-hoc approaches is performed via a small-scale workshop session.

The rest of this paper is structured as follows; Sect. 2 introduces our security process patterns and then Sect. 3 presents, via a working example, a framework that utilises them for the creation of secure business process designs. Section 4 presents the evaluation of the proposed set of security process patterns, while Sect. 5 compares the contributions of our work to related literature. Finally, Sect. 6 concludes with a short discussion of this work and its future directions.

2 Security Process Patterns

A pattern, in the context of software development, is a reusable package which incorporates expert knowledge and represents a recurring structure, activity, behaviour or design [22]. A security pattern is a well-understood solution to a recurring information security problem and can be expressed either as a structural pattern, which incorporates designs that can be implemented in the final product or a procedural pattern, which represent high level directions for improving the process of developing security-critical software systems [10]. During the requirements and analysis phases of the system development lifecycle, the majority of the proposed design pattern focus on security attacks while patterns for

implementing countermeasures are less represented [22]. Therefore, as part of this work we introduce a number of structural process design patterns aiming to model the implementation of countermeasures for the main types of security requirements at a business process model level of abstraction. Such patterns are generic enough to be implementation-agnostic but able to specify a basic sequence of activities and interactions between process participants which lead to the satisfaction of the system's security requirements.

The basic structure of each of the proposed patterns is captured using BPMN collaboration diagrams [18] and includes the activities required for the operationalisation of a security implementing technology, annotated with a pad-lock symbol at their top left corner to visually communicate their security-implementing nature. Corresponding activities exist at the user's lane describing any required interaction with the system's security implementing activities (e.g., username and password input). The security constraint activity or resource, which created the need for the implementation of security, is marked with a bold black border in order to be easily distinguishable. The activities contained within each pattern are not dependent on the implementation of a specific mechanism but rather on the type of the security requirement at hand. Therefore, the same pattern can be instantiated by a number of different mechanisms (e.g., smartcard, biometrics, username/password) which implement the same type of requirement (e.g., authentication). It is also the case that one pattern can be reused within another pattern depending on the security requirement it captures. For instance, the pattern for Authentication is reused within the Authorisation pattern since its functionality is required for the completion of the authorisation process. The patterns proposed by this work for each type of security requirement are the following:

Authentication. Authentication in the context of a business process requires a user to have a verified identity before performing a specific activity or accessing a resource. To realize the authentication requirement, as illustrated in Fig. 1, every time a user submits a request to the system for accessing an authentication-constraint resource or activity, the system should check that request and ask for the user's authentication data. Once the user submits the authentication data in the appropriate form (e.g., username/password, biometric data) the system should check its validity and if it is valid allow the user to access to the constraint resource or activity.

Authorisation. Authorisation, in terms of a business process model, requires that only users with the appropriate permissions can access a resource or perform an activity. As shown in Fig. 2, to realise the authorisation requirement, first a user requests access to authorisation-constraint activities or resources and the authentication process takes place in order for the user's identity to become known to the system. After the successful authentication, the role and/or the permissions attached to the user's account are checked and, if appropriate, the user gains access to the constraint activity or resource.

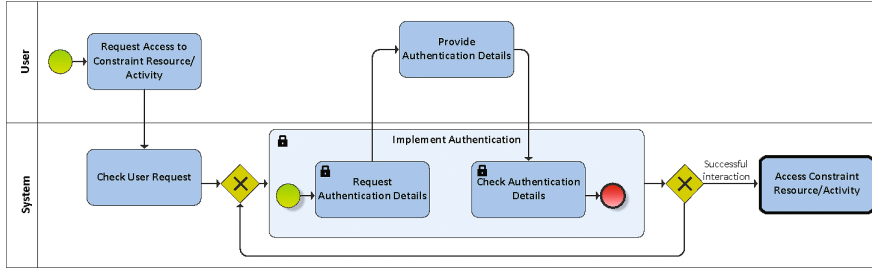


Fig. 1. Authentication pattern

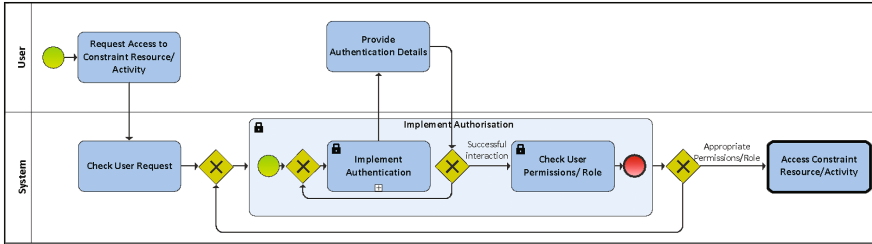


Fig. 2. Authorisation pattern

Confidentiality. Confidentiality ensures that any resource exchanged between the user and the system cannot be accessed by any unauthorised third party. As shown in Fig. 3, to achieve confidentiality in a business process, if the user is not already authorised, the authorisation process takes place as previously described. Next, a secure communication channel is created between the user and the system through which the confidentiality-constraint resource can be transferred.

Integrity. Integrity requires that resources exchanged between the user and the system cannot be modified during their transfer. As illustrated in Fig. 4, to achieve integrity, after an integrity-constraint resource has been transferred to the system, the system's copy of the resource needs to be compared to the original by data validation techniques.

Availability. Finally, the pattern for availability, presented in Fig. 5, is utilised to ensure that critical resources are always available to system users. To realise that requirement, when a requested resource is not available, the system has to maintain backups, using a number of available implementation technologies, from which the resource can be retrieved and be made available to the user.

3 Secure Business Process Design Framework

The process patterns introduced in the previous section are an important component of our ongoing work on the development of a secure business process

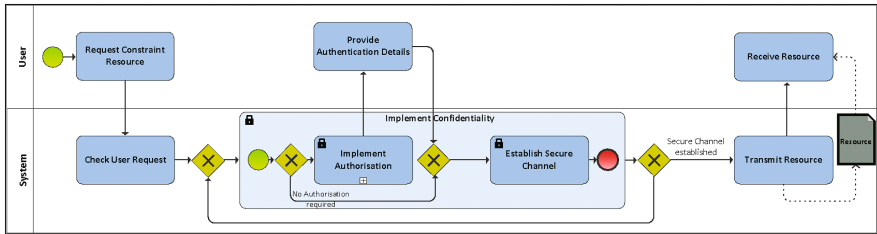


Fig. 3. Confidentiality pattern

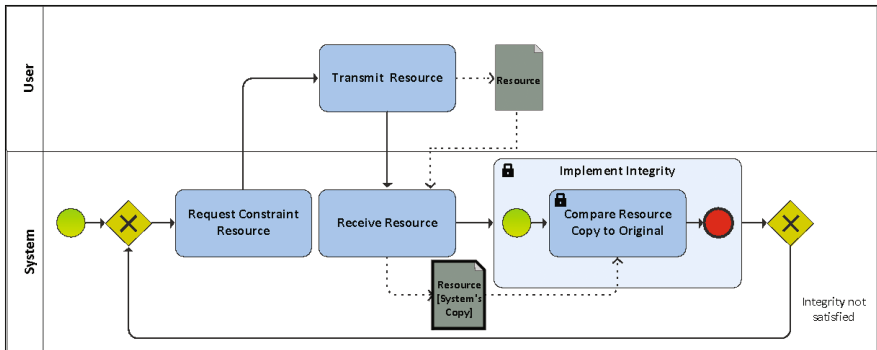


Fig. 4. Integrity pattern

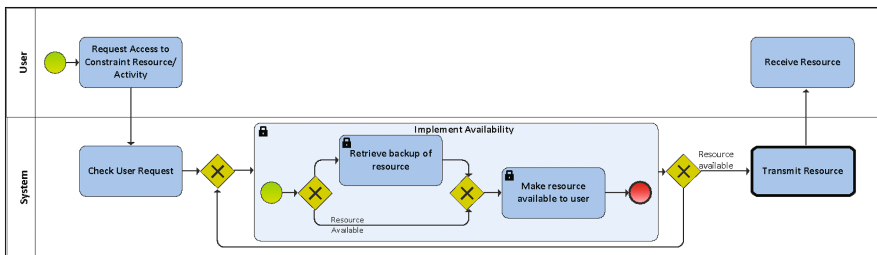


Fig. 5. Availability pattern

design framework [2–4]. The main objective of this framework is to create secure business process designs using as input the high-level security requirements of system stakeholders captured via organisational goal models. An illustration of the different components of the framework and their interconnections is provided in Fig. 6.

The steps for the application of our framework are described in Fig. 7. *Step 1* uses the *Goal Modelling component* to create a security-oriented goal model that captures a high abstraction view of the system to-be. Once such model has been created, a series of model transformation steps are applied in *Step 2* using the

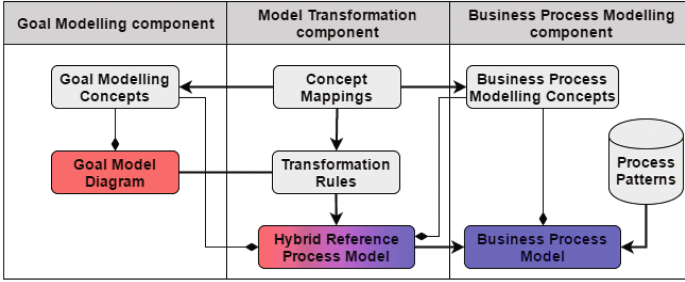


Fig. 6. Components of secure business process design framework

Model Transformation component to create a hybrid reference process model. That model acts as a mid-level artefact which maps security requirements and proposed countermeasures to specific parts of the business process, thus creating a security-annotated process skeleton. *Step 3* makes use of the *Business Process Modelling component* to refine the hybrid reference process model in order to create the final output of our framework, a secure business process model. During this step, the proposed security process patterns are integrated (*Step 3.1*) and instantiated (*Step 3.2*) in the final business process model and the process flow is manually determined (*Step 3.3*).

A more detailed overview of the framework’s application, incorporating the security patterns introduced in Sect. 2, will be demonstrated via a working example of an electronic prescription system. The purpose of that system is to facilitate the creation and archiving of electronic prescriptions created by medical practitioners and used by patients to receive medication. Through the application of our framework a number of models of this system will be created, each capturing a different level of abstraction, with the secure business process model of the e-prescription process being the final output.

Goal Modelling Component. The creation of security-oriented goal models for the elicitation of requirements, threats and potential implementation mechanisms for the system to-be is the starting point of our framework. The ability of Secure Tropos [14, 15] to capture and analyse such concepts in an explicit and structured manner is the main reason for its selection as the modelling language of choice for performing the organisational level modelling.

An example of a Secure Tropos goal model diagram is presented in Fig. 8. The entities interacting within that system, namely the “*E-prescription system*”, the “*Medical Practitioner*” and the “*Patient*” are represented as actors. Each of them has a set of goals to achieve by interacting with each other. Their goals are decomposed into sub-goals and finally into plans which represent simple activities each actor has to perform (e.g., “*Create new prescription*”). Resources are also identified to represent documents created or required by plans or goals in order to be fulfilled (e.g., “*Prescription*”). Security constraints are connected to goals, plans or resources in order to restrict their functionality in favour of

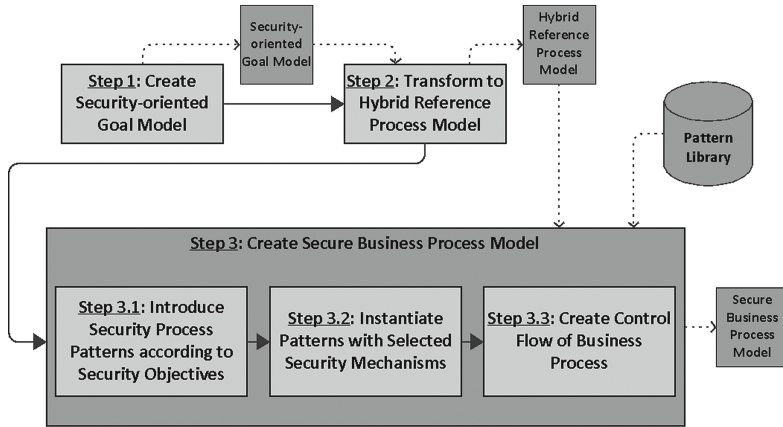


Fig. 7. Steps for the application of framework

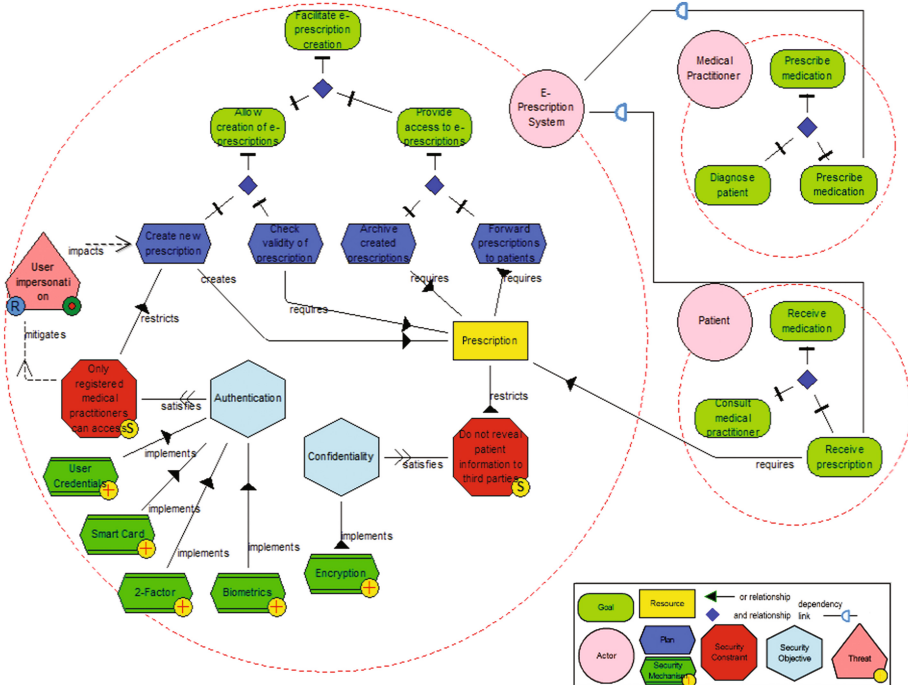


Fig. 8. Security Requirements view model of E-Prescription System

achieving a security objective. For instance, in the system modelled in Fig. 8, “Only registered medical practitioners can access” is an “Authentication” type constraint, while the “Do not reveal patient information to third parties” is a “Confidentiality” type of constraint. Threats (e.g., “User Impersonation”) are

also identified and connected to entities they can potentially impact. To achieve the system's security objectives and mitigate the identified threats, a number of security implementing mechanisms are introduced. For example the security objective of "*Authentication*" can be satisfied by the implementation of "*Two-step authentication*" or "*Smart Cards*". System designers and security experts are encouraged to propose any mechanism that may fit the needs of the system at this stage, since the final decision regarding the mechanisms that will be implemented in the final business process will take place at a later time.

Model Transformation Component. To achieve linkage between the goal model and the operational level of abstraction at which business processes operate, the model transformation component of our framework introduces an intermediate model called *hybrid reference process model*. It includes concepts from both goal and process models (therefore *hybrid*) and can capture the variability introduced by the different options regarding security implementing mechanisms, as previously identified at the goal model (therefore *reference model*). The process related concepts (i.e., lanes, activities, data objects) included in the hybrid reference process model are transformed from their corresponding goal model concepts (i.e., actors, goals, plans, resources). The security related information, captured by Secure Tropos concepts (i.e., constraints, objectives, mechanisms, threats) and connected to elements of the goal model, is transferred as-is to the equivalent concepts of the hybrid reference process model.

The application of the model transformation component at the e-prescription system's goal model produces the hybrid reference process model illustrated in Fig. 9. More specifically, the actors introduced during the organisational level analysis of the system (i.e., *Patient*, *Medical Practitioner* and *E-Prescription System*) are transformed into business process lanes with the same name. Next, activities are created and placed in the corresponding lanes, originating from the leaf-level goals and plans of each system actor. For instance the "*Diagnose Patient*" leaf-level goal is transformed into an activity with the same name in the *Medical Practitioner's* lane. In a similar manner, the relevant resources (i.e., *Prescription*), previously introduced at the goal model, result in data objects in the hybrid reference process model, connected as inputs or outputs to the activities that create or require them. For instance, since the "*Prescription*" resource is created by the plan "*Create New Prescription*" at the goal model level, a data resource with the same name is the output of the corresponding activity in the hybrid reference process model.

The constraints connected to a goal, plan or resource of the goal model are now transferred at the hybrid reference process model and connected to the corresponding activity or data object (i.e., "*Only registered medical practitioners can access*" connected to the "*Create new prescription*" activity). The security constraints, which are now linked with specific process elements, are connected to security objectives, transferred from the goal model (i.e., "*Authentication*"). The security objectives categorise the identified security constraints and also help the process designers to select the appropriate process pattern which will

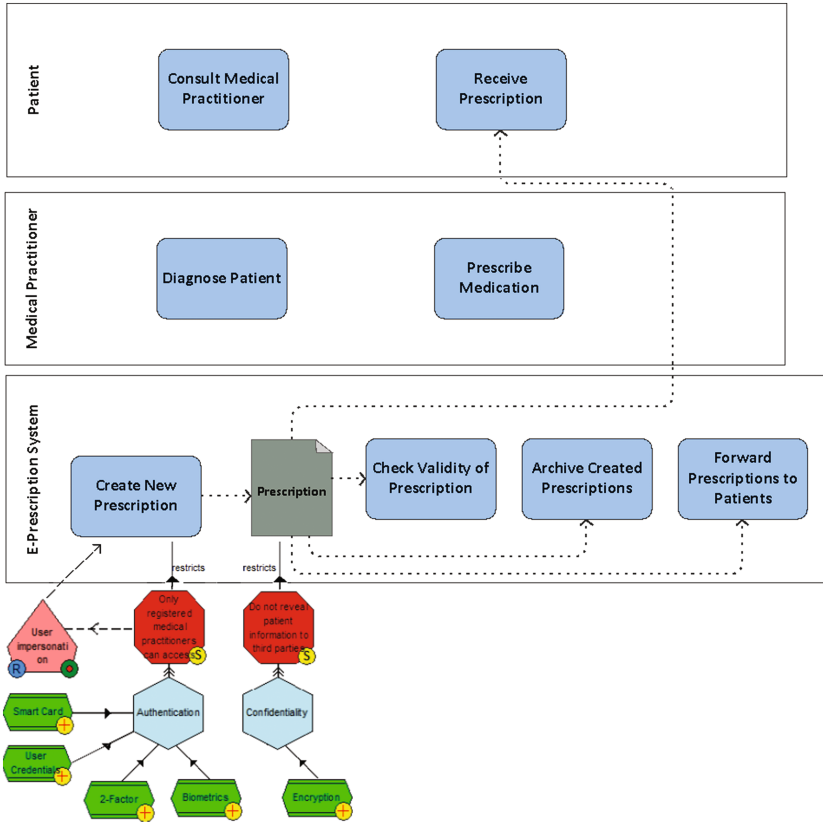


Fig. 9. Hybrid reference process model of the E-Prescription System

be integrated in the final business process model. The security mechanisms, proposed at the goal model for the implementation of each security objective, are also transferred in the hybrid reference process model to maintain the information regarding the range of potential configurations of security countermeasures at the process level. For instance, *Smart Cards*, *Biometrics* or *Usernames and Passwords* are amongst the security mechanisms that can be selected for the implementation of the *Authorisation* security objective, linked via the “*Only registered medical practitioners can access*” constraint to the “*Create new prescription*” activity.

Business Process Modelling Component. The business process modelling component uses the hybrid reference process model as input for creating secure business process designs. The security process patterns, introduced earlier in this work, are used at this point in order to guide the integration of the selected security mechanisms in the final business process model.

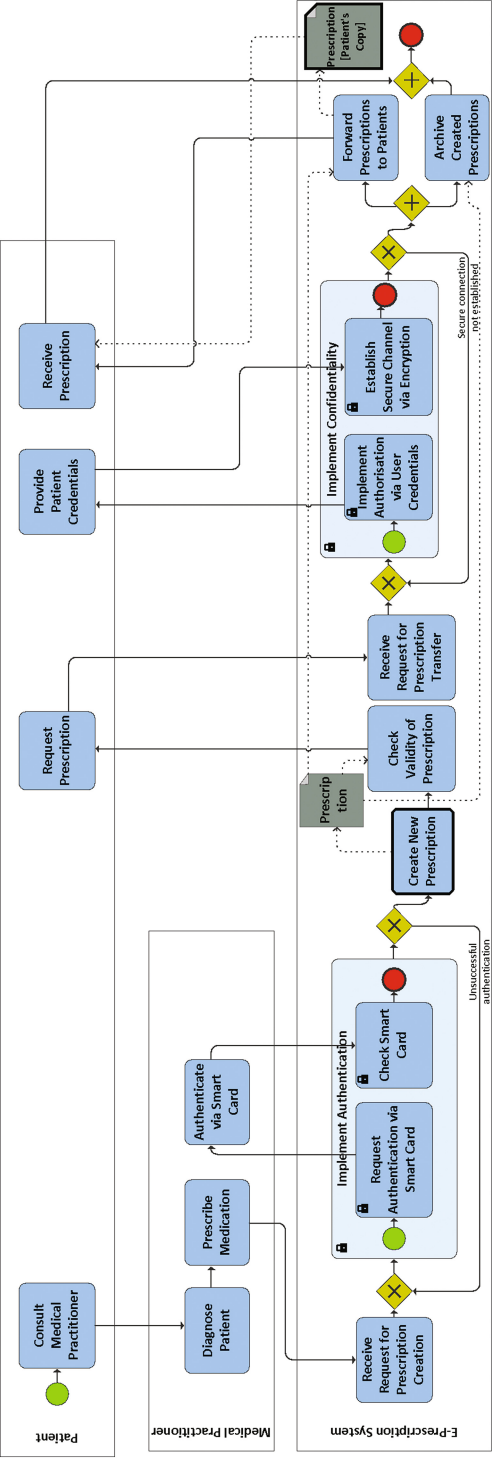


Fig. 10. Business process model of the E-Prescription System

Figure 10 presents the final business process model of the E-Prescription system. In the “*E-Prescription System*” lane of the business process model, the business process design pattern for the requirement of “*Authentication*” (c.f. Fig. 1), has been introduced before the security constraint activity “*Create new prescription*”, denoted with a bold-line border. The authentication pattern has been instantiated in order to implement the “*Smart Card*” security mechanism. Therefore, activities of the authentication pattern which were abstractly defined, such as “*Request Authentication Details*” are instantiated into more explicit declarations (i.e., “*Request Authentication via Smart Card*”) to reflect the implementation of the selected security mechanism. The same process was followed for the “*Confidentiality*” requirement connected to the “*Prescription*” resource, where the pattern for “*Confidentiality*” (c.f., Fig. 3) has been instantiated to implement the encryption security mechanism.

Other than the introduction of the instantiated process patterns, the control flow of the final business process has been manually created, including start and end events, gateways and flows indicating the order of execution of the included activities. The introduction of the control flow elements is a manual task since the goal model, which provided us with information regarding the basic structure of the intended system, is inherently not equipped to capture information regarding its temporal dimensions, such as the order of execution of its activities.

4 Evaluation

A small-scale experiment was conducted in order to (i) evaluate the perceived understandability and ease-of-use of the proposed security process patterns and (ii) compare their implementation to ad-hoc security integration in business process models. Twelve (12) postgraduate students (MSc and PhD level), in the areaS of information systems design and information security, completed the experiment.

The whole process was performed in a supervised workshop session with a total duration of thirty minutes. A brief overview of business process modelling concepts and diagrams was provided before the beginning of the experiment. After that, a fragment of the business process presented in Fig. 10 was presented to the participants without any security implementing activities. They were first asked to introduce any extra activities they considered necessary, in an ad-hoc manner, in order to satisfy the authentication objective for a specified element of the process. Only after that phase of the experiment was completed, the participants were presented with the authentication pattern. They were asked to apply it to the initial business process model fragment in order to accomplish the same security objective. After both phases were completed a short questionnaire was distributed in order to capture the opinions of the participants regarding their experience. The questionnaire entries were phrased as statements accompanied by a 5-point Likert scale, ranging from strongly disagree to strongly agree, from which the responders selected the option best reflecting their opinion. At the end of the questionnaire form there was also the option of providing free-form

comments and remarks¹. According to the participants' responses at the questionnaire:

- 83% either agreed or strongly agreed that the provided process pattern was easy to understand,
- 75% either agreed or strongly agreed that the provided process pattern was easy to integrate to the existing process model,
- 42% either agreed or strongly agreed that it was difficult to identify in an ad-hoc manner, the security related activities needed to be added in the process.
- 67% either agreed or strongly agreed that it easier to create a secure business process model using the provided process pattern compared to the ad-hoc security implementation.

The experiment allowed us to get an indication of the perceived usability and understandability of the proposed process patterns. It also indicated that such patterns are a preferable alternative to ad-hoc approaches as they provide more structure and guidance to process designers. A limitation of our experimental setup is its small sample size which limits the significance of its findings. The generalisability of the results is also limited since the participants only worked with a small subset of the proposed patterns. Nevertheless, the responses gathered through this small-scale experiment provide a valuable starting point for the further development and future evaluation of secure business process patterns.

5 Related Work

Kienzle et al. [10] have created a pattern repository including both structural and procedural patterns for web service security, expressed through a textual template. Mouratidis et al. [16] introduce security patterns to describe security implementing techniques (e.g., agent authenticator), expressed using Tropos, an agent-oriented software engineering approach, and a textual description template. Rosado et al. [19] link security requirements to architectural and design security patterns in order to guide the implementation of security in the area of web services. High-level architectural patterns and mid-level design patterns of security implementing mechanisms (e.g., secure message router, credential tokenizer) are matched to specific types of security requirements of web service applications. Ahmed et al. [1] identify potential risks and security requirements at the process level by matching process fragments with security-risk patterns used to capture common security requirements. A comprehensive survey of works in the area of security design patterns is provided by Laverdière et al. [11], where a number of desirable properties of security design patterns and a template for pattern description are developed.

The above works [10, 16, 19] provide patterns which aim to capture specific types of security countermeasures or, in the case of [1], use process patterns to

¹ The questionnaire and a summary of the responses can be accessed in: <http://www.sense-brighton.eu/process-patterns-questionnaire/>.

identify where security-related violations can occur within the process. Each of the patterns presented in our work captures the operationalisation of one type of security requirement and can accommodate its implementation by any suitable security implementing technology. Therefore, their implementation-independent nature, allows a higher degree of generalisability and flexibility compared to countermeasure-specific patterns.

Salnitri et al. [20] introduce SecBPMN which extends BPMN 2.0 in order to perform security-related annotation of business processes. The security requirements captured via such annotations are formalised by a series of predicates which, similar to security process patterns, encapsulate security-related information. Li et al. [13] introduce a method for constructing goal models which are able to capture and analyse attack patterns depending on the contextual environment of the system. Kalloniatis et al. [8,9] introduce the PriS framework for the design of privacy-aware processes, starting from goal models. A set of privacy process patterns are used by PriS for the incorporation of privacy requirements into business processes, which are refined and expressed in BPMN 2.0 in [3].

Similar to the works above, our framework also uses of goal models but it provides explicit steps for transitioning from them to the operational level of abstraction. Additionally, it allows the mapping of both security requirements and security countermeasures, captured at a high abstraction level, to specific business process elements. Therefore, via the use of security process patterns, it facilitates the alignment between security requirements at the organisational level and the operationalisation of security countermeasures at the process level.

6 Conclusion

Designing secure business processes can be a challenging endeavour since system developers often have limited knowledge regarding the analysis and implementation of security. Process patterns, encapsulating expert knowledge and proven solutions, can be a way to overcome the lack of security expertise during a system's development process. Identifying security process patterns of the appropriate abstraction level and granularity is another challenge, since over-specified patterns may be not flexible enough to fit the specific context of the system at hand, while high-level architectural patterns may be too generic.

The work presented in this paper proposes a series of reusable security-oriented process fragments which can be utilised as process patterns for the integration of security in business process models. This collection of patterns is used as a component of a broader framework for the design of secure business process models, the application of which has been illustrated through an example. The most important characteristic of the proposed process patterns is the level of abstraction at which they are expressed, as it allows them to capture the steps required for the operationalisation of security requirements in a generic but expressive and implementation-agnostic manner.

The perceived usability and understandability of the proposed patterns was positively evaluated during a small-scale workshop session. The participants of

the same workshop session also indicated that designing secure processes via the proposed set of patterns was preferable to ad-hoc approaches to security.

Our future work in this area will focus on the further refinement and extension of the proposed pattern library. In addition to that, the privacy process patterns, introduced by our previous work [3], will be added to the pattern library of our framework so it will be able to cover the analysis and operationalisation of both security and privacy countermeasures in business process models. Finally, a large-scale evaluation of the overall framework via a case study of an existing system will allow us to extract valuable conclusions regarding its applicability.

Acknowledgement. This research received funding from the Visual Privacy Management in User Centric Open Environments (VisiOn) project, supported by the EU Horizon 2020 programme, Grant agreement No 653642.

References

1. Ahmed, N., Matulevičius, R.: Securing business processes using security risk-oriented patterns. *Comput. Stand. Interfaces* **36**(4), 723–733 (2014)
2. Argyropoulos, N., Márquez Alcañiz, L., Mouratidis, H., Fish, A., Rosado, D.G., Guzmán, I.G.-R., Fernández-Medina, E.: Eliciting security requirements for business processes of legacy systems. In: Ralyté, J., España, S., Pastor, Ó. (eds.) *PoEM 2015. LNBP*, vol. 235, pp. 91–107. Springer, Cham (2015). doi:[10.1007/978-3-319-25897-3_7](https://doi.org/10.1007/978-3-319-25897-3_7)
3. Argyropoulos, N., Kalloniatis, C., Mouratidis, H., Fish, A.: Incorporating privacy patterns into semi-automatic business process derivation. In: *IEEE 10th International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–12. IEEE (2016)
4. Argyropoulos, N., Mouratidis, H., Fish, A.: Towards the derivation of secure business process designs. In: Jeusfeld, M.A., Karlapalem, K. (eds.) *ER 2015. LNCS*, vol. 9382, pp. 248–258. Springer, Cham (2015). doi:[10.1007/978-3-319-25747-1_25](https://doi.org/10.1007/978-3-319-25747-1_25)
5. Decreus, K., Poels, G.: A goal-oriented requirements engineering method for business processes. In: Soffer, P., Proper, E. (eds.) *CAiSE Forum 2010. LNBP*, vol. 72, pp. 29–43. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-17722-4_3](https://doi.org/10.1007/978-3-642-17722-4_3)
6. Decreus, K., Poels, G., Kharbili, M.E., Pulvermueller, E.: Policy-enabled goal-oriented requirements engineering for semantic business process management. *Int. J. Intell. Syst.* **25**(8), 784–812 (2010)
7. Fernandez, E.B., Pan, R.: A pattern language for security models. In: *Proceedings of PLoP*. vol. 1 (2001)
8. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Using privacy process patterns for incorporating privacy requirements into the system design process. In: *2nd International Conference on Availability, Reliability and Security (ARES 2007)*, pp. 1009–1017. IEEE (2007)
9. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the PriS method. *Requirements Eng.* **13**(3), 241–255 (2008)
10. Kienzle, D.M., Elder, M.C.: Security patterns for web application development. University of Virginia Technical report (2002)
11. Lavédière, M., Mourad, A., Hanna, A., Debbabi, M.: Security design patterns: Survey and evaluation. In: *2006 Canadian Conference on Electrical and Computer Engineering*, pp. 1605–1608. IEEE (2006)

12. Leitner, M., Miller, M., Rinderle-Ma, S.: An analysis and evaluation of security aspects in the business process model and notation. In: 8th International Conference on Availability, Reliability and Security (ARES 2013), pp. 262–267. IEEE (2013)
13. Li, T., Paja, E., Mylopoulos, J., Horkoff, J., Beckers, K.: Security attack analysis using attack patterns. In: IEEE 10th International Conference on Research Challenges in Information Science (RCIS), pp. 1–13. IEEE (2016)
14. Mouratidis, H., Argyropoulos, N., Shei, S.: Security requirements engineering for cloud computing: the Secure Tropos approach. In: Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.) *Domain-Specific Conceptual Modeling, Concepts, Methods and Tools*, pp. 357–380. Springer, Cham (2016)
15. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. *Int. J. Softw. Eng. Knowl. Eng.* **17**(2), 285–309 (2007)
16. Mouratidis, H., Weiss, M., Giorgini, P.: Modeling secure systems using an agent-oriented approach and security patterns. *Int. J. Softw. Eng. Knowl. Eng.* **16**(03), 471–498 (2006)
17. Neubauer, T., Klemen, M., Biffl, S.: Secure business process management: a roadmap. In: 1st International Conference on Availability, Reliability and Security (ARES 2006), p. 8. IEEE (2006)
18. Object Management Group: Business Process Model Notation (BPMN) Version 2.0. Technical report (2011)
19. Rosado, D.G., Gutiérrez, C., Fernández-Medina, E., Piattini, M.: Security patterns and requirements for internet-based applications. *Internet Res.* **16**(5), 519–536 (2006)
20. Salnitri, M., Dalpiaz, F., Giorgini, P.: Designing secure business processes with SecBPMN. *Softw. Syst. Model.*, 1–21 (2016)
21. Weske, M.: *Business Process Management: Concepts, Languages, Architectures*. Springer, Heidelberg (2010)
22. Yoshioka, N., Washizaki, H., Maruyama, K.: A survey on security patterns. *Prog. Inform.* **5**(5), 35–47 (2008)

Enterprise, Business-Process and Information Systems
Modeling

18th International Conference, BPMDS 2017, 22nd
International Conference, EMMSAD 2017, Held at CAiSE
2017, Essen, Germany, June 12-13, 2017, Proceedings
Reinhartz-Berger, I.; Gulden, J.; Nurcan, S.; Guédria, W.;
Bera, P. (Eds.)

2017, XV, 355 p. 105 illus., Softcover

ISBN: 978-3-319-59465-1