Integrating security patterns into the electronic invoicing process

Michael Netter and Günther Pernul
Department for Information Systems
University of Regensburg
Regensburg, Germany
{*michael.netter, guenther.pernul*}*@wiwi.uni-regensburg.de*

***Abstract*—The increasing automation of business processes is one of the main benefits of the ongoing technological evolution. Regarding e-invoices this automation process is still not optimally supported despite the fact that recent studies indicate a high potential to save costs. Within this paper we identify the main obstacles and propose a multi-stage solution. Therein we classify the e-invoicing process using common security objectives and, since the process includes many security related elements, propose an initial solution based on security patterns. The approach takes advantage of the main benefits of security patterns to provide a domain-independent solution which is built upon expert knowledge.**

***Keywords*-e-invoice; security patterns; security objectives;**

I. Introduction

Invoice processing is a central part of the value chain of almost any business. Therefore improvements may both have a positive financial and performance impact. For example, a recent study indicates a potential for savings up to 70 percent for e-invoices [1] compared to paper-based invoices. Basically every invoice (e.g. in PDF format) sent by electronic means may be considered an e-invoice. However there are several legal, mental and technical obstacles that make e-invoice processing a complex task.

According to [2] there are at least three main barriers which hamper the comprehensive adoption of e-invoices:

- **Legal uncertainty:** For instance, until recently there was no legal basis for e-invoicing in the European Union [3]. Legislation amendments in many states addressed this issue. However there is still a lot of uncertainty due to the fact that requirements of multiple legal domains overlap and need to be fulfilled.
- **Lack of trust:** The lack of trust in e-business solutions becomes apparent by comparing the requirements for paper-based and e-invoices. While the authenticity of origin of paper-based invoices is often solely verified on the basis of the letterhead, e-invoices mostly require an advanced or qualified electronic signature, which offers a significantly higher trustworthiness than paper invoices.
- **Missing standardisation:** The multitude of available specifications - partly proprietary - hampers both the interoperability when exchanging e-invoices and the establishment of an accepted standard. This applies especially for cross-nation interactions, where different legal requirements have to be considered.

It is evident that developing a consistent e-invoice solution is a complex task. We propose a multi-stage process to resolve the afore-mentioned obstacles. This includes the partitioning of the e-invoice process and the classification according to the core security objectives. Subsequently security patterns are assigned to each objective. Using security patterns, we aim to overcome each of the difficulties. By providing a domain-independent solution, security patterns can help to construct a solution that abstracts from legal requirements of a particular legal system. Furthermore a solution built upon security patterns may be more trustworthy since security patterns embody expert knowledge and usually patterns went through a public verification process. Finally, the fact that security patterns are based on well-proven, field-tested solutions may be beneficial for standardisation. It has been demonstrated by Fernandez et al. [4] that business solutions may benefit from a pattern based approach.

The remainder of this paper is structured as follows. Within section II we analyse the shortcomings of paper-based invoices and outline a solution for e-invoicing. Preliminary work for our approach is presented in section III and IV. In section V we demonstrate the integration of security patterns into the e-invoicing process. Section VI describes related work and section VII concludes the paper with an outlook on future work.

II. Problem statement and proposed solution

Despite the possibility of sending e-invoices many enterprises still rely on paper-based solutions. The default paper-based invoicing process is depicted in figure 1. Thus the invoice is created for instance using an ERP-Software like

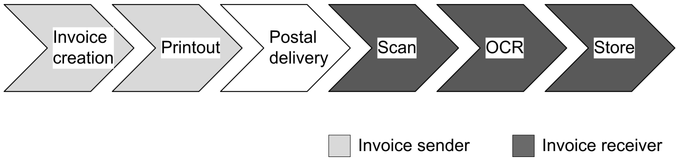

Figure 1. Default paper-based invoicing process

SAP and afterwards printed on paper. After postal delivery the receiver scans the invoice to re-digitalise it and uses an OCR-Software to extract relevant information. Usually both the paper-based invoice and the digitalised invoice are stored.

It is obvious that this solution has several shortcomings. First both printing and scanning cause media breaks which are prone to error since the invoice layout is not standardised and therfore it is hard to automatically extract all the relevant details using OCR-Software. Furthermore this solution increases costs by providing the hardware infrastructure and by having to pay for postal delivery. Additionally sending the invoice by mail is much more time-consuming than sending it by electronic means.

To overcome these shortcomings and to arrive at an interoperable secure e-invoice we propose a multi-stage solution which is depicted in figure 2. Therein the first steps are to partition the e-invoicing process into logical units and to select appropriate security objectives from an eligible security standard (e.g. ISO 27001:2005 [5]). Afterwards we apply the selected security objectives to the partitioned e-invoicing process in order to get a classification scheme. This is used to examine which security objectives are relevant for each unit of the partitioned e-invoice process.

As explained in section I, our solution is based on security patterns. Therefore it is necessary to classify the security patterns using the afore-mentioned security objectives. This step is essential to combine the e-invoicing process with security patterns. When integrating both concepts, each security objective in the classification scheme that is required for a process unit is replaced by appropriate security patterns that fulfil this objective. Furthermore this extended classification scheme enables us to derive a pattern language (see figure 2), i.e. a consistent solution that illustrates not only the security patterns used but also the relations between those patterns. However due to space constraints those two final steps are not presented in this paper and are left out to future work.

III. E-INVOICE PROCESS PARTITIONING AND CLASSIFICATION

The paper was motivated by the fact that the default e-invoice process is a complex task that is hampered by several obstacles. Since it is easier to address subproblems, simplifying the e-invoicing task is essential for the development of a consistent solution. Therefore we propose to partition the process at a high level of abstraction as follows.

- **Creation:** Refers to the process of creating a valid e-invoice. This includes measures to ensure a secure, auditable creation process as well as the protection of the integrity and the authenticity of the origin of an e-invoice.
- **Exchange:** Refers to the process of sending, transmitting and receiving e-invoices and the affected core security objectives.

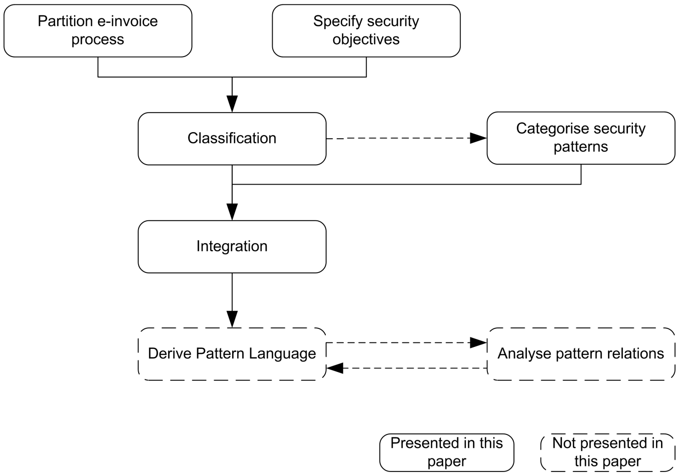
Partition e-invoice process
Specify security objectives
Classification
Categorise security patterns
Integration
Derive Pattern Language
Analyse pattern relations
Presented in this paper
Not presented in this paper

Figure 2. Approach of integrating security patterns into the e-invoicing process

- **Storage:** Refers to the process of persistently storing an e-invoice and all attachments that are required to verify the integrity and the authenticity of the origin of an e-invoice.

Since this is a rather high-level classification, the security objectives assigned to one process unit may also have an impact on parts of further process units. For instance, accountability requirements may apply to the creation process and furthermore when sending an invoice which is part of the exchange process, while accountability may not be required for the transmission.

After partitioning the e-invoice process we propose to classify each group to refine the security requirements and subsequently being able to assign security patterns to each requirement. This paper focuses on addressing security related issues of the e-invoicing process and therefore we propose the classification following the ISO 27001:2005 standard [5].

Table I illustrates the classified e-invoicing process. Thus the e-invoice creation has to fulfil several requirements. Authenticity must be ensured (e.g. by signing the invoice using an electronic signature). Furthermore integrity needs to be guaranteed to prevent - accidentally or intentionally - modified documents to be sealed and sent. Additionally accountability is needed to meet legal requirements.

The exchange process can be classified as follows. If the e-invoice is transmitted using an insecure or public communication channel (e.g. the internet) confidentiality may be endangered. Furthermore appropriate measures must be applied to be able to ensure integrity. It is also necessary to verify the authenticity of the document and to be able to proof the origin of a message (non-repudiation).

Because of legal requirements each invoice must be kept for a specific period of time, for example ten years in Germany. This implies that for the storage process, several

	Creation	Exchange (send, transmit, receive, verify)	Storage
Confidentiality		(x)	(x)
Integrity	x	x	x
Availability			x
Authenticity	x	x	x
Non-repudiation		x	
Accountability	x		(x)

x: is required (x): may be required

Table I
E-INVOICE PROCESS CLASSIFICATION

security objectives are relevant. First an enterprise may want to ensure the confidentiality of the stored invoices. Furthermore it is important to ensure integrity of the stored documents (e.g. e-invoice and related signature) for the whole storage period. Due to legal requirements it is also necessary to maintain the availability of the storage system, e.g. for VAT audits. These requirements also imply that the authenticity of an invoice needs to be verifiable for the whole storage period. Furthermore it may be necessary to consider accountability requirements.

IV. CLASSIFYING SECURITY PATTERNS USING SECURITY OBJECTIVES

To combine security patterns with the e-invoicing process, it is required to classify security patterns using the same classifiers as applied to the invoicing process. The classification scheme developed in section III provides the basis for the integration of security patterns. Security patterns usually contribute to fulfil one security objective. They may however influence the achievement of other security objectives, but analysing these side effects is out of scope of this paper and part of our future work.

The first step is to analyse the body of security patterns and to categorise the relevant patterns according to the security objectives determined in section III. However this is not a trivial task because there are several books [6][7] and multiple repositories to screen for relevant security patterns and a complete catalogue is still missing. Furthermore the common templates (e.g. the POSA [8] and the GOF [9] template), that are used to structure patterns, lack an indication of the security objectives a pattern addresses. To resolve these shortcomings, several approaches have been published. In [10] the development of a wiki based pattern catalogue is proposed to ease the identification of relevant patterns. In order to resolve the problem of relating security patterns to security objectives Yskout et al. [11] propose to extend existing pattern templates to include information about affected security objectives.

On the analogy of [12] we surveyed 215 security patterns published in 36 pattern catalogues. For classification, we adapted the scheme proposed in [11] to fit our needs. Furthermore, few core security objectives have been combined to more generic classes, since several security patterns address a more general security objective that can hardly be mapped to only one security objective (e.g. the *single sign-on delegator* pattern [6] is beneficial for both the identification, authentication and authorisation process as well for the availability of those services). The consequence of this classification is that some categories overlap and dependencies emerge. Those effects have also been recognized in [11].

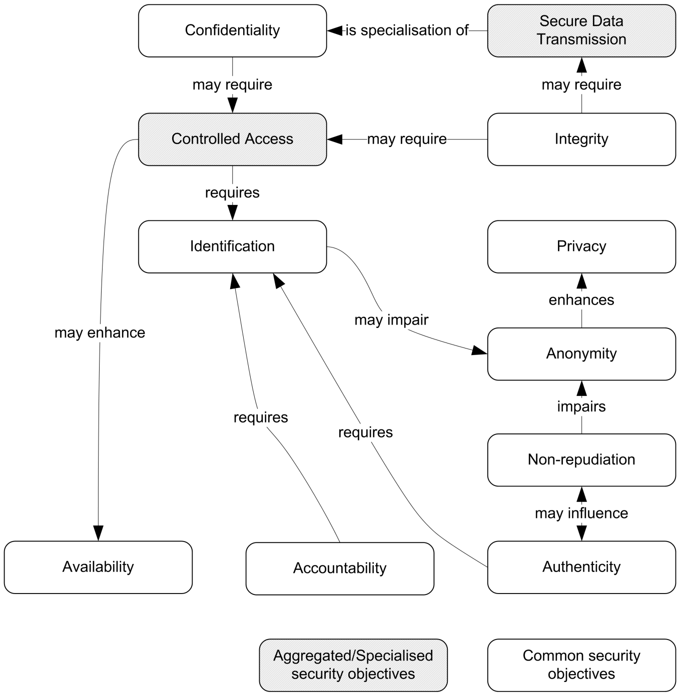

Figure 3. Security pattern classification scheme following [11]

Figure 3 illustrates the classification scheme. To better fit the security patterns' orientation, the scheme consists of an ”aggregated” objective (Controlled Access), a ”specialised” objective (Secure Data Transmission) and the common security objectives. The fact, that the scheme contains more security objectives than used in section III to classify the e-invoicing process is because invoicing does not affect all objectives (e.g. privacy) but a complete scheme was required to classify all security patterns. The relations between those objectives are also depicted in figure 3. Those relations become useful when deriving a pattern language for e-invoicing since a pattern fulfilling one objective may have a positive or negative effect on the adaptability of patterns from another category.

V. INTEGRATING SECURITY OBJECTIVES AND SECURITY PATTERNS

After the preliminary work in section III and section IV, this section describes the integration of security patterns into the e-invoicing process.

To map security patterns to a process unit (e.g. exchange) and a security objective (e.g. integrity) it is required to create a set of security patterns that contribute to fulfil this security objective and carefully review all patterns within this set. However due to the generic character of security objectives, only some of the patterns of this set are applicable while the others fulfil the security objective within another context. For instance, the *Virtual Address Space Access Control* pattern [13] is both beneficial for integrity and confidentiality but applies to a different domain (operating systems) and is therefore not adaptable to ensure the integrity or the confidentiality of an e-invoice.

Table II illustrates the integration of security patterns into the e-invoicing process. Due to space constraints we have exemplary chosen three fields from the whole classification matrix. The first field contains security patterns that are beneficial for the integrity within the creation process of an e-invoice. Due to the generic classification scheme, the integrity can refer to different domains within the creation process (e.g. it may refer to the process of ensuring the integrity of the invoice to avoid signing accidentally or intentionally modified documents but it may also refer to the integrity of the signature creation device). Therefore in table II there are multiple patterns which contribute to fulfil the security objective integrity.

The same applies for accountability within the e-invoice storage process. While some of the patterns refer to the accountability of an incident to a system (*secure logger* [6]), others aim to improve the accountability of an incident to a person (*share responsibility for security* [14]). Furthermore it can be seen that there are overlapping patterns. For instance, the *log for audit* [14] and the *three-point logging* [15] pattern have a similar intention, which requires to choose one alternative when deriving a pattern language.

The assignment process also revealed a strong relation between non-repudiation and authenticity. If a security pattern ensures the authentiticy of a message this often implies that the non-repudiation requirements are also fulfilled. For example, the *sender authentication* [16] pattern requires to sign a message using a private key. Verifying this signature using the public key implies the authentiticy of the message and makes it impossible for the sender to deny the authorship of the message.

There are serveral notable points that became evident when assigning security patterns to the e-invoicing process. First each field of the classification matrix contains multiple security patterns. This enables us to keep the solution independent from a concrete implementation. For instance, there are patterns that ensure integrity using a modification detection code while others rely on electronic signature. A decision for a concrete implementation has to be made when deriving a pattern language for e-invoicing using this classification scheme.

Process: ***Creation*** **Security objective:** ***Integrity***
Authorative Source Of Data
Client Input Filters
Input Guard
Message Integrity
Certificate Authority
Certificate Revocation
Key In The Pocket

Process: ***Storage*** **Security objective:** ***Accountability***
Continual Status Reporting
Three-point Logging
Audit Interceptor
Log For Audit
Secure Logger
Share Responsibility For Security

Process: ***Exchange*** **Security objective:** ***Non-repudiation***
Secrecy With Signature
Message Authentication
Sender Authentication

Table II
INTEGRATION OF SECURITY PATTERNS INTO THE CLASSIFICATION SCHEME

Furthermore there were a few fields in the matrix where we could hardly find appropriate security patterns but only a few that address only parts of the problem or contribute to fulfil a security objective as a side effect while having its main focus on another objective. This is a strong indicator for missing patterns. For example, most patterns within the storage process that contribute to the security objective integrity relate to the integrity of the system but it is hard to find appropriate patterns for message integrity. Moreover in some cases it is necessary to combine several security patterns to fulfil a security objective. Accountability is a good example for this case. To implement a comprehensive logging solution both the *audit interceptor* [6] pattern and the *secure logger* [6] pattern are required. While the former enables us to be noticed of all important events the later provides functionality for storing the information in a way so that it cannot be modified.

VI. Related Work

Having a suitable classification scheme is essential to find the right pattern. While we used security objectives for the classification, several other schemes have been proposed. Most of them are based on manual classification (see [17] for a comprehensive overview), however there are approaches to classify patterns automatically [18] [19].

Using security patterns to develop secure applications is an active research area. Several approaches have been proposed to integrate security patterns into the development lifecycle. [20] presents an approach which is based on an UML extension, called UMLSec. Braz et al. propose an approach which is based on threat modeling and misuse activities [21].

VII. Conclusion

Within this paper we identified the main obstacles that hamper a comprehensive application of e-invoices. We proposed a solution based on a security objective classification scheme that takes advantage of the domain-independence and the expert knowledge of security patterns. Therein we demonstrated that a combination of security patterns and security objectives proposed by Yskout et al. [11] is possible.

However the classification scheme has a few shortcomings mostly due to its generic nature. Therefore our future work includes the analysis of more fine-grained classification schemes that create smaller categories and hence enable a better assignment of security patterns. This also facilitates a better identification of issues that are not yet covered by appropriate security patterns. We also plan to integrate threat analyis and misuse cases to further refine out approach [21]. Futhermore we aim to derive a pattern language from the classification scheme to show a concrete solution for the e-invoicing process. This also includes analysing the dependencies and relations between patterns since the application of one security pattern may hamper the applicability of another pattern.

Acknowledgment

This work is partly funded by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology and carried out within the SecPat project (Contract No. IUK292/002).

References

[1] PricewaterhouseCoopers, "Study on the requirements imposed by the member states, for the purpose of charging taxes, for invoices produced by electronic or other means," 1999. [Online]. Available: http://ec.europa.eu/taxation_customs/resources/documents/taxation/vat/key_documents/reports_published/TenderXXI-98-CB-5010.pdf

[2] European Commission Informal Task Force on e-Invoicing, "European e-invoicing final report," 2007. [Online]. Available: http://ec.europa.eu/information_society/eeurope/i2010/docs/studies/eei-3.2-e-invoicing_final_report.pdf

[3] European Council, "Amendment to directive 77/388/EEC with a view to simplifying, modernising and harmonising the conditions laid down for invoicing in respect of value added tax, directive 2001/115/EC," 2001.

[4] E. B. Fernandez and Y. Liu, "The account analysis pattern," in *Proceedings of the European Conference on Pattern Languages of Programs Conference*, 2002.

[5] ISO 27001:2005, *Information technology – Security techniques – Information security management systems – Requirements*. ISO, Geneva, Switzerland, 2005.

[6] C. Steel, R. Nagappan, and R. Lai, *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*. Prentice Hall, 2005.

[7] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns : Integrating Security and Systems Engineering*. Wiley & Sons, 2006.

[8] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*, 1st ed. Wiley & Sons, 1996.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

[10] S. Henninger and V. Corrêa, "Software pattern communities: Current practices and challenges," in *Proceedings of the Pattern Languages of Programs Conference*, 2007.

[11] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen, "A system of security patterns," Katholieke Universiteit Leuven Department of Computer Science, Tech. Rep., 2006.

[12] T. Heyman, K. Yskout, R. Scandariato, and W. Joosen, "An analysis of the security patterns landscape," in *Proceedings of the Third International Workshop on Software Engineering for Secure Systems*. IEEE Computer Society, 2007.

[13] E. B. Fernandez, "Patterns for operating systems access control," in *Proceedings of the Pattern Languages of Programs Conference*, 2002.

[14] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt, "Security patterns repository version 1.0," 2002. [Online]. Available: http://www.scrypt.net/ celer/securitypatterns/repository.pdf

[15] P. Dyson and A. Longshaw, "Patterns for managing internet-technology systems," in *Proceedings of the European Conference on Pattern Languages of Programs Conference*, 2003.

[16] A. M. Braga, C. M. F. Rubira, and R. Dahab, "Tropyc: A pattern language for cryptographic software," in *Proceedings of the Pattern Languages of Programs Conference*, 1998.

[17] M. Hafiz and R. Johnson, "Security patterns and their classification schemes," University of Illinois at Urbana-Champaign Department of Computer Science, Tech. Rep., 2006.

[18] A. Kubo, H. Washizaki, and Y. Fukazawa, "Extracting relations among security patterns," in *Proceedings of the Second Workshop on Software Patterns and Quality*, 2007.

[19] A. Kubo, H. Washizaki, A. Takasu, and Y. Fukazawa, "Analyzing relations among software patterns based on document similarity," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2005.

[20] J. Jürjens, *Secure Systems Development with UML*. Springer, 2004.

[21] F. A. Braz, E. B. Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities," in *Proceedings of the Second Workshop on Secure Systems Methodologies using Patterns*, 2008.