SEPA Files Transmission: Implementing Security Guarantees in Enterprise Resource Planning Systems

Diogo Gonçalves[1] and Isabel Seruca[2,3]
[1]SBX Consulting, Rua Gonçalo Cristovão, 347- MAPFRE Building, room 207, Porto, Portugal
[2]Univ. Portucalense, Research on Economics, Management and Information Technologies - REMIT, Rua Dr. António Bernardino Almeida, 541-619, P 4200-072, Porto, Portugal
[3]ISTTOS, Centro Algoritmi, University of Minho, Portugal
diogo.mg25014@gmail.com, iseruca@upt.pt

Keywords: SEPA, ERP, Security, Encryption, Hashing.

Abstract: The SEPA regulation has defined a set of technical and business requirements and common standards that any payment system must respect to be considered compatible with the Single Euro Payments Area (SEPA) project. The technical requirements and the mandatory nature set by the EU of joining the SEPA project require a set of adaptations to be made by companies in their business relationship with Payment Service Providers (PSPs), with particular emphasis on: adapting their Enterprise Resource Planning Systems (ERPs), often referred to as "ERP SEPA compliance", and the integration of secure C2B file transmission solutions, since a XML file is readable and editable. This paper describes a project developed at SBX Consulting targeting the implementation of security guarantees for the sending of SEPA files between a client company and the banking entities with which the company works. The security software solution developed addresses the encryption and hashing of the SEPA files and was integrated into the existing SAP system used by the company.

1 INTRODUCTION

As a natural consequence of the creation of the single currency, the Single Euro Payments Area (SEPA) was created in 1999 by the European Commission, the Eurosystem and the Banking Sector in Europe with the aim of strengthening the European integration with the establishment of a single market for retail payments (EPC, 2017). With SEPA, all retail payments in Euros are considered "domestic", thus leaving no differentiation between international and domestic payments when made within the Eurozone and acceding countries.

SEPA (EPC, 2017) is a geographic space where individuals, companies and public administration can make and receive payments in euros, under the same conditions, rights and obligations within the Euro Zone, regardless of their location. Under SEPA, payment instruments, such as credit transfers, direct debits and payment cards, are used identically in all participating banks, whether domestic or cross-border transactions are considered, with only one bank account.

The benefits associated with the use of SEPA are consensually recognized (Barbas, 2009; Lloyds Bank, 2017; Harsink, 2010; HSBC,2017): (i) All euro payments made through a bank adhering to SEPA can be made with the same bank account and costs at which national payments are made; (ii) Provides greater protection to users of payment services (payment services directive 2007/64/EC of 13th november 2017); (iii) Defines common rules and standards, contributing to a better efficiency in the execution of payments; (iv) Centralization of treasury management, saving time and costs; (v) Payment management in the SEPA space is facilitated by centralizing transactions in a single account and using the same format for all incoming and outgoing payments.

The regulation of the SEPA initiative, through regulation (EU) No 260/2012 of the European Parliament and of the European Council of 14 March (EUR-Lex, 2017), imposed a mandatory obligation to join the project for companies and payment service providers (PSPs), setting a deadline (1 February 2014) for the coexistence of national direct transfer and debit systems and SEPA systems. This legal

Gonçalves, D. and Seruca, I.
SEPA Files Transmission: Implementing Security Guarantees in Enterprise Resource Planning Systems.
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018) - Volume 1*, pages 205-212
ISBN: 978-989-758-298-1

framework focuses on a set of technical and business requirements and common standards for credit transfers and direct debits in euros, covering the interbank relationship, but also issues of the relationship between banks and their customers.

One of the technical requirements specified in Regulation 260/2012 is the mandatory use of (batch) instructions based on the ISO 20022 XML format, both in the relationship between Banks and in their relationship with Business Customers (including micro-enterprises).

The SEPA Payments and Transfers service is currently offered by Banks to Business Customers (companies), through the sending of SEPA C2B (Customer-to-Bank) files in ISO 20022 XML format. It is common for companies to use this service by pooling a set of credit transfers in Euros, for their own countries and the SEPA space, in a single file in XML standard format (for example, by loading a batch of Payment Orders related to the payment of salaries).

The technical requirements and the mandatory obligation of joining the SEPA project require a set of adaptations to be made by Companies, with particular emphasis on:

(i) adaptation of their enterprise resource planning (ERPs), often referred to as "ERP SEPA compliance" (ING Belgium SA, 2013; Barbas, 2009), in particular with regard to the cash management module and in the ability to generate files in XML format for SEPA transactions, manipulate mandatory data formats such as IBAN and BIC and generate SEPA specific reports;

(ii) integrate solutions to send C2B files safely, since an XML file is readable and manipulable.

The first issue has been addressed by most enterprise management software vendors by updating and extending the commercial versions of the major ERPs (SAP, 2006; MS Dynamics Nav, 2016; SAGE, 2017; Primavera, 2014, PHC, 2014). The second issue has been mostly addressed from the point of view of the communication service offered by banking institutions with special emphasis on the security connection. However, in addition to the connection being secure, it is important to ensure the confidentiality and the integrity of the data exchanged between a company and a PSP, particularly in the case of sensitive information.

The work described in this paper addresses this last issue and arose from a project developed at SBX Consulting (SBX, 2017) for the implementation of security guarantees in the sending of SEPA files between a client company and the banking entities with which the company works. It was also requested to integrate this solution into the existing SAP system used in the management area of the company.

The project explores the design and development of a software application integrated into the SAP platform with two main components: (i) the hashing component for the implementation of the integrity security guarantee developed in Advanced Business Application Programming (ABAP) and (ii) the encryption component for implementing the confidentiality security guarantee, implemented through the GnuPG application.

This paper is structured as follows: Section 2 presents the design and rationale for the architecture of the software solution to be developed. In Section 3 the development of the CryptoSafe application is described, identifying the main types of processes to be supported, and presenting the system modeling and the interface considered. Finally, Section 4 concludes with some considerations on the main challenges addressed, the current stage of the project and future steps to be undertaken.

2 SOLUTION DESIGN

2.1 Envisaging the Solution

As a result of the internal audit performed on the client company where the project was to be implemented, and since the SAP system was already used in the management of the company, it was decided that the following software would be used in the implementation of the solution:

- the open-source GnuPG (GPG) software (GnuPG, 2017), which provides tools to create keys, encrypt and decrypt files;
- SAP ABAP functions to perform the hashing.

Figure 1: Integration of the application with SAP and GnuPG.

In order to address the two issues of the problem - (a) encryption, so as to implement the confidentiality guarantee and (b) hashing, so as to implement the integrity guarantee - the operation of the two programs (GnuPG and SAP) was integrated in the application (cf. Figure 1), while hashing was implemented in SAP ABAP.

The application was termed “CryptoSafe”, alluding to the purpose of the software and resulting from the junction of the English words “Cryptography” and “Safe” from Safety deposit box.

The use of GPG requires the execution of commands of the following type:

```
gpg [options] [file_name]
```

ABAP is the programming language used to develop code in SAP. Based on the study performed, the following method was used for hashing calculation and employed in the application coding:

```
cl_abap_message_digest=>calculate_hash_for_char
```

2.2 Related Work

In the literature review performed, we tried to identify tools and applications that could solve part or the whole process of encryption and hashing, allowing integration with ERP systems.

Although no products were found that did exactly the same as the software application to be developed, other SAP programs/applications that use PGP encryption were found, with special emphasis on Advantco's "PGP Solution" for SAP NetWeaver (Advantco International, 2017). The Advantco company provides support and implementation of encryption solutions for SAP NetWeaver. This software supports several encryption algorithms, namely RSA, Elgamal and DAS, which CryptoSafe will also use.

SAP NetWeaver is the technology platform for the latest SAP products released, including the application server (ABAP + JAVA stack), while SAP R3 is the conventional SAP ERP product before being replaced by SAP ECC (which is based on NetWeaver technology).

The advantage of CryptoSafe is that it can be used in both SAP products, unlike the solution marketed by Advantco.

2.3 Target Users

CryptoSafe is intended to be used by any company that, in the scope of its activity, needs to send SEPA files, while its use may be generalized to the context of sending sensitive or confidential information with the implementation of security guarantees.

The application also targets SAP users who need to save files in a secure way, while SAP developers have access to the encryption/hashing tools in the source code.

3 CRYPTOSAFE DEVELOPMENT

3.1 Identification of Types of Processes

Within the context of the CryptoSafe application, “processes” are pre-written tasks that may run directly without the need to re-enter all the data. Each process can be saved and executed only for a given function type; the possible types of actions and subsequent actions are:

- Encryption;
- Decryption;
- Sign;
- Unsign;
- Create Hash;
- Compare Hash:
  - Compare normal file with the file that has the hash;
  - Compare normal file with the string where the hash is inserted;
- Import Public Key;
- Import Private Key;
- Export Public Key;
- Export Private Key.

3.2 Modelling of the System

Figure 2 shows the Use Case diagram related to the system. The diagram depicts three types of users that can interact with the CryptoSafe system:

- the developer who can only access the advanced low level functions (Encryption, Decryption, Sign Document, Unsign Document, Import Public/Private Key, Export Public/Private Key, Create Hash, Compare Hashes, Manage Keys), being able to execute these functions separately;
- the SAP technician with permission to create processes of Encryption, Hashing, Signing (associating the respective algorithms and keys in the process fields and in the applicable cases), edition and deletion of processes, being also able to perform the functions of the developer and access the low level functions;
- the manager who has access to process selection and execution functionalities, besides having permissions to the profiles of SAP developer and SAP technician.

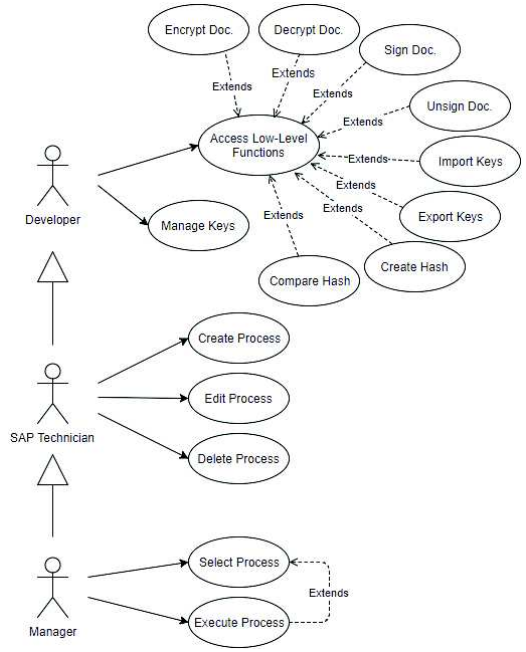

Figure 2: Use Case Diagram for CryptoSafe.

The Entity Relationship Diagram for the system is shown in Figure 3.

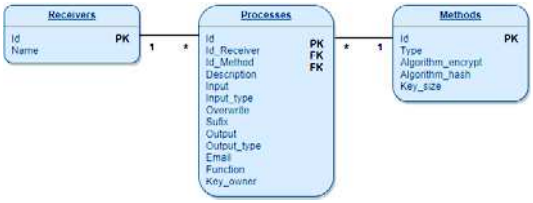

Figure 3: ER Diagram for CryptoSafe.

3.3 Developing the GPG Commands

As already stated, the GNU Privacy Guard (GnuPG or GPG) software tool provides several symmetric and asymmetric encryption features. In order to use the tool within the context of CryptoSafe development, a study of its operation and syntax was performed. When using the tool through the command console, the syntax is the following:

gpg [options] [file_name]

Before any action may be taken, a key is needed in the GPG system; this key can be created using the command:

gpg --gen-key

When executing the command in the console, several questions are presented to the user with multiple choice responses on the command line, namely the type of algorithm used, key size, the name of the key, etc. The user, by choosing the algorithm to create the key, will define whether the algorithm will be used only for signing and unsigning documents, or whether it will be used for encryption, decryption, signing, and unsigning.

Choosing the algorithm will also set the maximum bit size the key may have, ranging from 3072 to 4096 bits. Next, the user will be asked to define the key's lifetime; this can last for days, weeks, months, years or simply have no expiration date. The user will also be prompted to provide the key ID, which will be used for identification beyond his fingerprint. The key ID is composed of:

Key name; Key email; Key comments.

After entering these data, GPG asks for a password setting for the key, displaying its "strength". The key is created and stored in the GPG key list and can be immediately used.

To browse all the keys contained in GPG, the following command should be used:

gpg --list-keys

After executing this command, all keys and corresponding parameters will be displayed, according to the output shown in Figure 4.

pub [public key size] / [fingerprt abrev.] [creation date]

uid [trust level] [name] [comments] [email]

sub [private key size] / [fingerprt abrev.] [creation date]

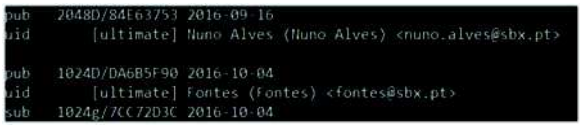

Figure 4: Keys presented by GPG (test).

Within the scope of this project, there was a concern to avoid to the maximum the direct contact of the user with the console of GPG, thus avoiding possible errors in the system. To this end, the feature of creating keys from ".txt" files was investigated, that is, the creation of keys without user interaction directly in the console, as long as those files respected certain syntax rules. This process is explained below:

Considering a document named "text.txt" and with the following contents:

```
Key-Type: 1
Key-Length: 2048
Subkey-Type: 1
Subkey-Length: 2048
Name-Real: Root Superuser
Name-Email: rot@handbook.westarete.com
Name-Comment: Superuser's key
```

```
Expire-Date: 0
Passphrase: password123
```

The following gpg command should run to create the key using the "text.txt" file:

```
gpg --batch --gen-key text.txt
```

When running the command and "pointing" to the file with the parameters specified in the "text.txt" document, the system automatically creates a key. This information became extremely relevant as it allowed the creation of keys without direct interaction in the console by the user.

The remaining commands for exporting and importing keys, encryption and decryption of files and signature and unsignature of files were studied and tested as well; however, for reasons of space limitation are not exemplified here.

3.4 CryptoSafe Architecture

According to the Use Case Diagram shown in Figure 2, three entities were considered to interact with the application, with different accesses to the system functionalities.

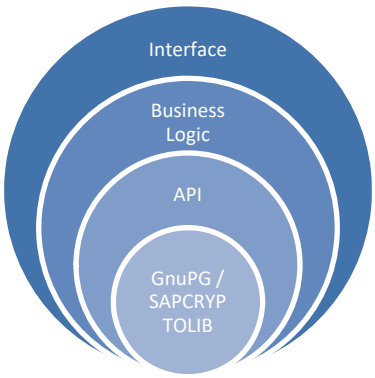

Figure 5: Layers structure of the CryptoSafe application.

Thus, the application was structured in four layers, illustrated in Figure 5 and described as follows:

GnuPG/SAPCRYPTOLIB - This layer consists of the GPG program and its encryption features; it also contains the SAP package named "SAPCRYPTOLIB" which includes the functions used by SAP to create and compare hashes.

API - The API layer contains the functions of the program at the low nomenclature level; this layer is accessed by the developer, who can perform the advanced functions of the program namely encrypt, decrypt, etc., being also responsible for key management.

Business Logic - It is considered the "soul of business" layer. At this layer, the SAP Technician can create, edit, and delete processes that are stored on the system, as well as have access to the developer layer and the lower layers, if desired.

Interface – It is the layer for the use of the manager; here, the manager can access all the features available at the lower levels, including those of the SAP technician and developer, as well as being able to select and execute processes.

The layers were designed to facilitate the understanding of the program functioning, making its manipulation by the user more intuitive and user-friendly. As the program evolves from the higher layer (Interface) to the lower layers, its operation becomes more "low level", and making more difficult to a user with less knowledge of how the program or the concepts of encryption work, to perform any type of operation, which could easily run from the Interface.

3.5 CryptoSafe Interface

According to the purpose of the application, the various interfaces were developed taking into account who would access them and the main functions performed.

Figure 6 shows the first screen after running the program. The only actor who must be able to act at this point in the execution of the application is the Manager.

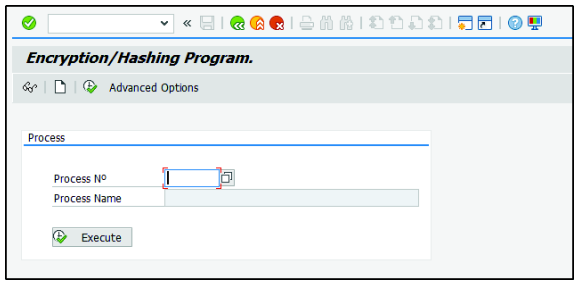

Figure 6: CryptoSafe application high-level screen.

By selecting one of the existing processes, the process can run directly or a new input and output path can be added (Figure 7). If these parameters are filled, the process will be executed taking into account the new input and output data for the files, using the methods (in this case encryption) recorded in the process when executing.

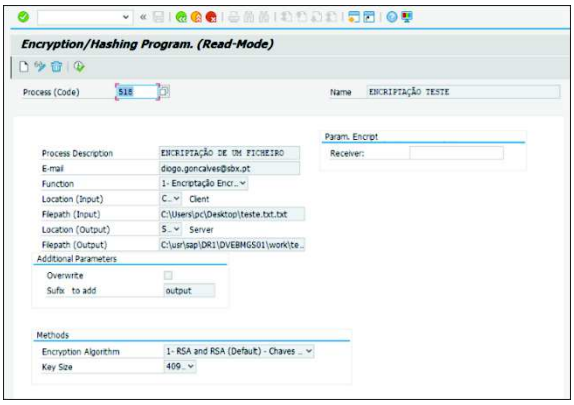

Figure 7: High-level screen with a process filled.

Figure 8 shows the mid-level interface, which can be accessed by the SAP technician and the Manager.

This interface presents all the data that constitute a process.

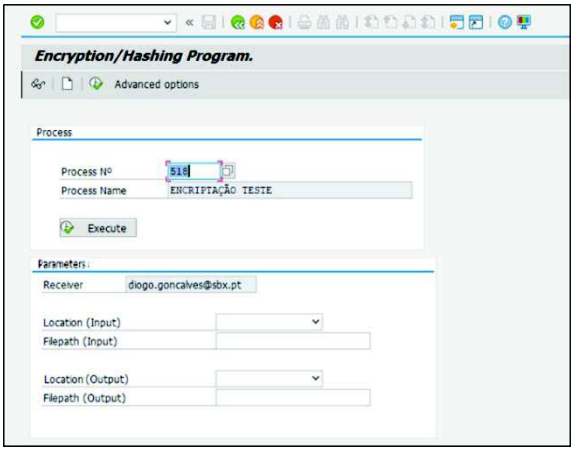

Figure 8: CryptoSafe application mid-level screen.

Finally, the low-level interface (Figure 9) is presented, for which the Developer is responsible, with the role of managing the keys that will be used in the GPG and the low-level functions. This interface can be accessed by any user; the execution of any operation requires the insertion of all requested data.

For the sake of illustration, an example of a SEPA file is shown in Figure 10 and its corresponding encrypted contents by using CryptoSafe and AES 256-bit encryption is shown in Figure 11.

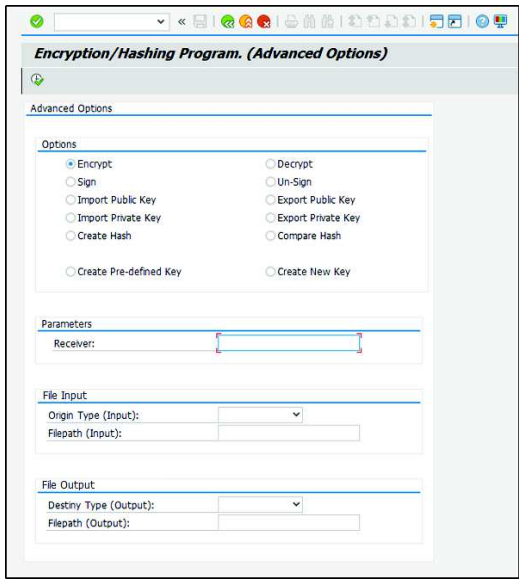

Figure 9: CryptoSafe application low-level screen.

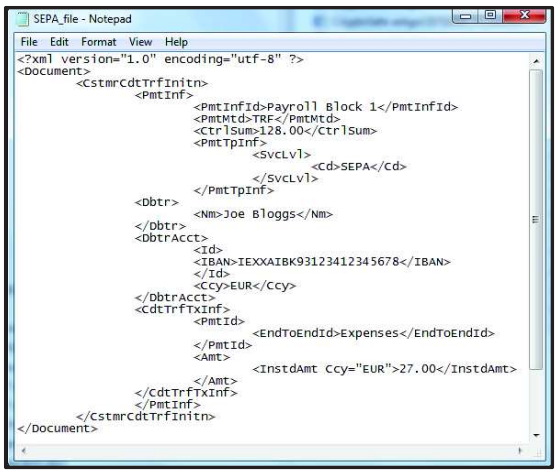

Figure 10: Example of a SEPA file.

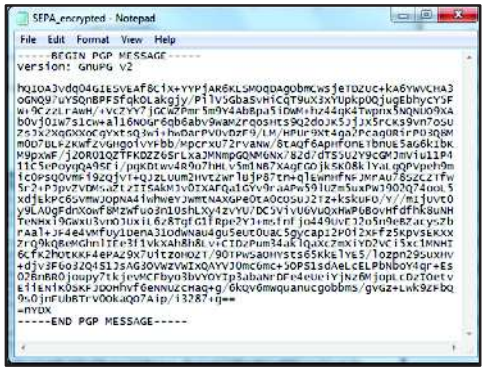

Figure 11: Encrypted contents of the SEPA file shown in figure 10.

4 CONCLUSION AND FURTHER WORK

As experiences and lessons learned with the development of this project, the following challenges related to the implementation are highlighted:

Interaction of the two programs - the implementation of this project required the communication between two distinct programs (ABAP and GnuPG) to address the hashing and the encryption/decryption of files, which implied the addressing of several issues, from security, to the implementation of the functionalities so as to provide a good experience to the end user.

The management of the executed requests was made with ABAP, while the answers were given by a program that is not integrated in ABAP (GnuPG); the requests had to be fulfilled and responded to effectively, in order to provide a satisfactory experience to the end user. To that end, ABAP was integrated with GnuPG, so that the former could execute certain GPG commands through the operating system on which the SAP server runs; in turn, it was necessary to configure them in the SAP program itself so that they could be called/executed using ABAP.

The SAP program has a transaction called "sm69", which allows to run external commands at the level of the operating system in which the program is located. By executing these commands, with GPG installed on the server and setting the correct parameters, GPG program operations may be executed, without forcing the user to have a direct interaction with the GPG console. This was the solution found for the implementation of encryption.

GPG Commands - GnuPG may not be considered a hard program to use; however, it is only user-friendly when the console is in front of the user and the user knows the syntax of the commands and which commands to use to implement the desired action. This was what was intended to be avoided in the implementation of CryptoSafe, that is, commands were developed, fixed and tested several times, in order to find out which commands the ABAP application should execute.

Overwrite and adding suffixes - Another problem of the program was that, when saving the files on the server, the file explorer on the server side did not allow the insertion of a new name and only assumed the name of files that already existed, that is, these would be replaced when the process would be carried out. In order to avoid this situation, a "fail safe" system was developed that allows the addition of suffixes at the end of the file, preventing accidental replacement. If the process included saving files on the client, then when executed more than once without changing the destination, it also replaced the file with the suffix; thus, an algorithm was developed that verifies the existence of the final file with a suffix; if that happens, it adds a number to the file and saves it, without replacing the file with the suffix.

Interface / Interactivity - Another issue that had to be considered was the tuning between the user and the application. The layout and its behavior were carefully studied in order to guarantee the user the most feasible usability. By using prototypes and screen layouts, the designed interfaces were tested by potential users, so as to ensure that their development took the end user to a proper course. In some way, a graphical user interface has been developed for the end user, thus avoiding direct interaction with GnuPG.

Finally, although the CryptoSafe application has already been targeted by a series of tests, it is not yet in operation in the Client Company, since there are other add-ons (e.g. human resource management improvements) and system upgrades that need to be approved in order to be implemented in conjunction with CryptoSafe.

At the time of the implementation, the Client Company and the banks, besides having a VPN connection between their private networks, will need to have the CryptoSafe application integrated in their SAP business models. Therefore, the security of the connection is reinforced with the implementation of security guarantees (confidentiality and integrity) of the SEPA files sent.

It is worth noticing that the commercialization of the CryptoSafe application is foreseen for other companies interested in acquiring the software to improve the security of the organization in the transmission of data.

ACKNOWLEDGMENTS

This research contribution was supported by SBX Consulting company and Portucalense University.

We thank, in particular, Luis Fontes from SBX Consulting (Luisfontes101@gmail.com) for his assistance in the field of encryption and IT Security along with his comments that greatly improved the manuscript.

REFERENCES

Advantco International, 2017. “Advantco PGP Solution for SAP Netweaver”, [Online], Available: https://www.advantco.com/product/solution/pgp [Accessed 20-Feb-2017].

Barbas, J. C., 2009. "The Single Euro Payments Area: A strategic business opportunity", Journal of Corporate Treasury Management, vol. 2, no.3, pp. 246-251, 2009.

EPC, 2017. European Payments Council (EPC), SEPA - Vision and Goals, [Online], Available: http://www.europeanpaymentscouncil.eu/index.cfm/about-sepa/sepa-vision-and-goals/ [Accessed 26-Nov-2017].

EUR-Lex, 2017. Regulation (EU) No 260/2012 of the European Parliament and of the Council of 14 March 2012 establishing technical and business requirements for credit transfers and direct debits in euro and amending Regulation (EC) No 924/2009, [Online], Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2012.094.01.0038.01.ENG&toc=OJ:L:2012:094:TOC [Accessed 15-Feb-2017].

GnuPG, 2017. “The GNU Privacy Guard”, [Online], Available: https://gnupg.org/ [Accessed 20-Feb-2017].

Hartsink, G. B. J., 2010. “Setting a deadline for migration to SEPA ensures planning security for all stakeholders”, Journal of Corporate Treasury Management, 4(1), pp. 35-45.

HSBC, 2017, “The benefits of SEPA, HSBC Global Banking and Markets” [Online], Available: http://www.hsbcnet.com/gbm/products-services/transaction-banking/payments-cash-management/europe/single-euro-payments-area/benefits.html [Accessed 30-Nov-2017].

ING Belgium SA, 2013. "ERP SEPA readiness checklist", version July 2013, [Online], Available: https://www.ing.be/static/legacy/SiteCollectionDocuments/ERP_SEPA_EN.pdf [Accessed 20-Nov-2017].

Lloyds Bank, 2017, “SEPA Direct Debit”, [Online], Available: https://www.bancobic.pt/img/21/201312_sepa.pdf https://commercialbanking.lloydsbank.com/products-and-services/cash-management/sepa-direct-debit/ [Accessed 20-Feb-2017].

MS Dynamics Nav, 2016. "How to: Set Up SEPA Direct Debit"[Online], Available: https://msdn.microsoft.com/en-us/library/dn414575(v=nav.90).aspx [Accessed 30-Oct-2017].

PHC, 2014. PHC Gestão CS, "O que muda no PHC CS com a SEPA?", [Online], Available: http://www.phc.pt/portal/e/sepacs.aspx [Accessed 20-Feb-2017].

PRIMAVERA, 2014. PRIMAVERA Business Software Solutions, SEPA - Single Euro Payments Area, Questões Frequentes, version 06.01.2014, [Online], Available: http://www.primaverabss.com/pt/UserFiles/Downloads/Quest%C3%B5esFrequentes_SEPA_08_01_2014%20PT.pdf [Accessed 20-Feb-2017].

SAGE, 2017, "Introduction to SEPA in Sage 200 Extra" [Online], Available: http://ask.sage.co.uk/scripts/ask.cfg/php.exe/enduser/std_adp.php?p_faqid=31365 [Accessed 20-Feb-2017].

SAP, 2006, "SAP redesigns ERP package for integration with SwiftNet and SEPA compliance" [Online], Available: https://www.finextra.com/newsarticle/15988/sap-redesigns-erp-package-for-integration-with-swiftnet-and-sepa-compliance [Accessed 20-Feb-2017].

SBX Consulting, 2017. SBX Consulting [Online], Available: http://www.sbx.pt/pt/empresa [Accessed 20-Feb-2017].