Validation of the electronic reports issued by the electronic fiscal cash registers ExportValidator application

Paul-Cristian VASILE

National Institute for Research and Development in Informatics – ICI Bucharest

paul.vasile@ici.ro

Abstract: With the change of the legislation in Romania, regarding the electronic fiscal cash registers, businesses have to equip themselves with devices that export the daily sales summary digitally, on an external storage device. The exported file must follow a well-defined structure and ensure data integrity by applying an included digital signature. Starting with 2018, the National Institute for Research and Development in Informatics – ICI Bucharest runs the technical certifying procedures for fiscal electronic equipment, being appointed by law as the certifying body. This involves verifying each cash register model marketed in Romania in order to issue the technical certification. Due to the high volume of devices submitted for testing, a manual check of the exported files requires both a long processing time and a broad knowledge of the cryptographic mechanisms involved in the digital signature procedure. In this context, the ExportValidator application was developed in order to automate the testing process and diminish the risk of human errors as much as possible. In this paper the application will be showcased, discussing some implementation details and some of the technologies used.

Keywords: electronic fiscal cash registers, validation, testing, digital signature, cryptographic mechanisms.

Validarea rapoartelor electronice emise de casele fiscale de marcat Aplicația ExportValidator

Rezumat: Odată cu modificarea legislației din România privind aparatele de marcat electronice fiscale, opera-torii economici sunt obligați să se echipeze cu dispozitive care exportă rezumatul zilnic al vânzărilor digital, pe un dispozitiv de stocare extern. Fișierul raportat trebuie să urmeze o structură bine definită și să asigure integritatea datelor prin aplicarea unei semnături digitale incluse. Din anul 2018, Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București a început aplicarea procedurii de aprobare tehnică pentru aparatele de marcat electronice fiscale, așa cum este prevăzut în lege. Aceasta implică verificarea fiecărui model de casă comercializată în România pentru a elibera avizul tehnic favorabil. Din cauza volumului mare de dispozitive trimise pentru testare, o verificare manuală a fișierelor exportate necesită atât un timp prelungit de procesare, cât și o cunoaștere aprofundată a mecanismelor criptografice implicate în procedura de semnare digitală. În acest context, aplicația ExportValidator a fost dezvoltată pentru a automatiza fluxul testării interne și pentru a limita cât mai mult posibil erorile umane. Această lucrare își propune prezentarea aplicației, a mecanismelor de verificare, a detaliilor de implementare și a tehnologiilor utilizate.

Cuvinte cheie: case de marcat electronice fiscale, validare, testare, semnătură digitală, mecanisme criptografice.

1. Introduction

The law regulating the validation and use of the new electronic fiscal equipment was amended in 2018. According to the law, the fiscal equipment must export the fiscal data of interest such as total sales, total VAT etc. in a digitally signed XML format file. Introduction of the obligation to use cash registers with electronic journal has been determined mainly by the need to:

- fight, as a matter of urgency, tax evasion; ensure a better monitoring on activities of economic operators, both retail or services providers;
- establish some levers in order to improve collection of budget revenues and reduce tax evasion;

https://doi.org/10.33436/v30i4y202009

- facilitate fiscal recording both for businesses and authorities, and better control of fiscal flows;
- stimulate the competitiveness in an honest business environment.

The fiscal cash registers in use until now in Romania were equipped with a roller paper-journal, and generated inconveniences both for retailers and for fiscal authorities. On one hand, the retailers need to purchase log rolls and spare parts used by manufacturers in EU countries, manufacturers which no longer mass-produce these types of devices with double rolls; on the other hand these devices do not allow monitoring of fiscal information or creating a risk analysis tool needed by fiscal authorities (Barbu D. C., 2016).

As a result, the legal framework was amended and supplemented, and now retailers and service providers have the obligation to use electronic fiscal equipment that electronically generate log files (so called "new generation cash registers", as defined in Art. 3, par. (2) of the Government Emergency Ordinance no. 28/1999, republished, amended and subsequently completed), as follows:

- starting with June 1, 2018, the businesses in the category of large and medium taxpayers, according to legal provisions;
- starting with August 1, 2018, the businesses in the category of small taxpayers, according to legal provisions.

Replacement of old generation cash registers with the new generation cash registers can be done in two ways:

- adaption of the old models, when possible, complying with the legal provisions in force;
- purchase of "new generation" equipment.

In this paper, the emphasis is on presenting the development stages of the ExportValidator application. Also, the methods and technologies used to complete the IT product are highlighted, and some use cases are presented.

2. Overview

ExportValidator is an application that facilitates testing of electronic fiscal cash registers by automatically performing the steps of the technical approval procedure. The application must allow verifying the integrity of the report file by validating the digital signature, it must ensure the XML conformity as defined by the National Agency for Fiscal Administration, and verify the accuracy of the reported fiscal values, according to a test configured previously and store in a database the history of the tests performed. The application is strictly intended to be used within the National Institute for Research and Development in Informatics by authorized personnel only and is accessed through a web interface on an address within the internal network of the institution. When a technician is authorized to run tests, the application administrator sets up an account on the test platform. The technician changes later the password generated by default. In the current configuration, the program runs on a "Private Cloud" type infrastructure, in the dedicated sector of the project "Cloud-type Infrastructure for Public Institutions in Romania – ICIPRO", using a virtual machine with the following allocated resources:

- 8 CPU cores;
- 16 GB of RAM memory;
- 120 GB of Storage.

The need for a high number of processing cores and volatile RAM capacity is due to ensuring the ability to support many simultaneous test sessions. Increased storage capacity is required by the internal procedure, as it states that the tested files must be permanently stored in their unaltered form (Zamfiroiu, Cirnu, Boncea, Rotuna, & Anghel, 2015). The file classes to be stored are as folows:

- qualified digital certificate (Used for digital signing);
- the signed message register file (Indicates the period for which the report was generated);
- fiscal report signed file (Contains the XML file, which includes the taxes, and digital signature);
- XML file extracted from the signed message (XML file in raw format).

3. Stages of development

Until reaching the final version, the ExportValidator application has known three stages of development.

The first version of validation - OpenSSL library

The first version of the report validator was built as a bash script, executable on a Linux platform. It used basic functions of the OpenSSL cryptographic library in order to load the file into the memory and verify the validity of the digital signature. OpenSSL is currently the most popular open-source library in the field of cryptography. It provides an impressive suite for software developers, established algorithms, and facilitates the work of the programmer up to an API level. It is available under the Apache license, which makes it suitable to use for both commercial and non-commercial purposes, as is the case with this application. Its open-source character ensures the constant contribution of the community, which leads to a rapid detection and resolution of possible performance or security vulnerabilities.

This validation variant implements the parsing of the structure of an X509 v3 digital certificate and the verification algorithm of the digital signature that uses the 2040-bit RSA algorithm for encrypting and the SHA256 algorithm for hashing.

The main disadvantages of the test module presented above are:

- the need for ICI Bucharest technicians to understand the library and the cryptographic mechanism (the need of personnel with thorough knowledge of cryptographic mechanisms);
- dependence to the Linux platform;
- lack of an efficient versioning system (Every technician could have a different version and that could lead to inconsistency of tests);
- absence of XML structure verification modules and reported values.

Thus, dealing from these disadvantages, we developed a second version of the tool, aiming to provide technicians with a user-friendly graphical interface that enables them to access three functional modules: digital signature validation, validating of the XML structure against an XSD schema, and reading the contents of a report extracted from a cash register.

The second version of validation - Java Desktop Application

For the development of the second version of the ExportValidator, the following technologies and libraries were used:

- the Java programming language;
- Java FX platform;
- the BouncyCastle cryptographic library.

Java is a high-level, object-oriented programming language invented by James Gosling and launched in 1995 by Sun Microsystems. The general purpose of the language was to give the programmer the ability to write software applications which can be run on any platform without the need to recompile the source code. To accomplish this, programs developed in Java are compiled

into bytecode, a specialized code that can run on Java Virtual Machines installed on various hardware architectures (Burd, 2012).

JavaFX is a Java-based graphical interface development library that gives programmers the ability to design, implement and test client-side applications that run on various platforms.

BouncyCastle is a collection of cryptographic APIs, developed for Java and C# languages. As of November 2016, they have FIPS 140-2 Level 1 certification, making them suitable for deployments in business-critical applications (Castle, 2016).

The second version of the application was created Using Java and JavaFX. It developed into a desktop application with a graphical interface that offered technicians functionalities such as: verifying the structure of a digital certificate, verifying the digital signature, extracting data from an XML file and using XSD schema in validating the structure of an XML report.

The transition from the OpenSSL script to the Java application produced a major improvement in the actual test time and a considerable decrease in the level of difficulty encountered by the test technician. However, many shortcomings of the first version were also found in the second version (Groza, 2012).

The main disadvantages of the second version were:

- lack of an efficient versioning system (Every technician could have a different version and that could lead to inconsistency of test);
- inability to verify the reported values.

The third version of the ExportValidator application, currently in use, aims to address these shortcomings by implementing the existing modules into an online platform that gives technicians the possibility to perform the tests as a web service. This was possible due to the availability of the BouncyCastle library for both Java and C# programming languages (Panda & Nag, 2015).

The third version of the application - ASP.net MVC web technology

The version currently used by ICI Bucharest technical staff for running the compliance tests on electronic fiscal cash registers is a WEB application, developed using the C# language and the ASP.net MVC web application development framework, from Microsoft. This technology implements the Model-View-Controller framework and is distributed under an open-source license, as opposed to the previous version, ASP.NET Web Forms, registered as proprietary software. The MVC architecture is used in software engineering and involves the separation of the business logic towards the user interface elements. Thus, any change that occurs on the visual aspect or logic does not affect the other levels of the application (Liberty & Hurwitz, 2003).

The Model-View-Controller architecture addresses two main objectives:

- parallel development – due to separation of components, separate teams of programmers can implement different parts of the application, without influencing each other. So, the development of an application can be divided into the development of the user interface (Frontend) and the development of the server component (Backend).
- code reuse – user interface components can be easily transferred to another application, even if it relies on a different data model (Albahari & Albahari, 2012).

Components of the MVC model

- **Model** – it is the independent part of the user interface, dealing with operations, logic and data management within the application, encapsulating classes of entities, giving the application an easy to understand and manageable format (Joydip, 2009).
- **View** – represents the user interface, has the role of displaying the received input from other components of the application.

- **Controller** – represents the interface between Model and View components, deals with processing the business logic and the requests received. It will detect all interactions and information entered in the View component, and then notify the Model component to manage the database (Figure 1).

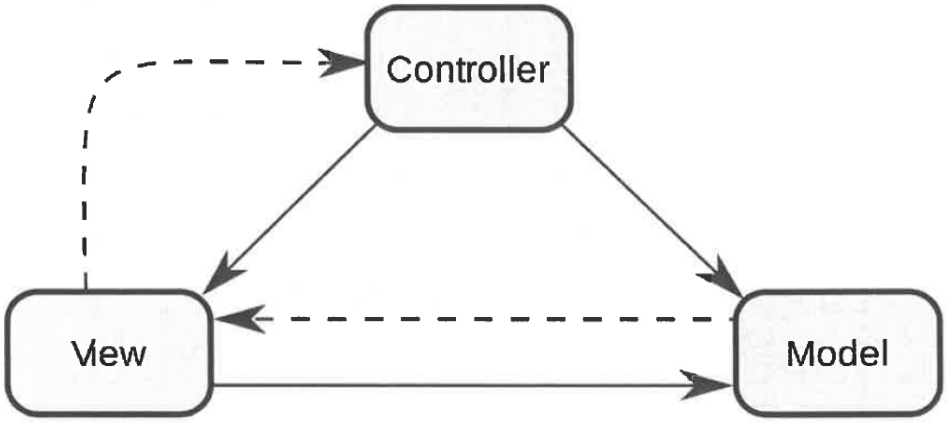

Figure 1. Architectural Model MVC [https://ro.wikipedia.org/wiki/Model-view-controller]

- Router - deals with determining, based on a URL received from the web browser, which method the controller will run. Upon receiving the request, the router will choose the appropriate route from an associated route table, then call the appropriate action of a controller (Figure 2).

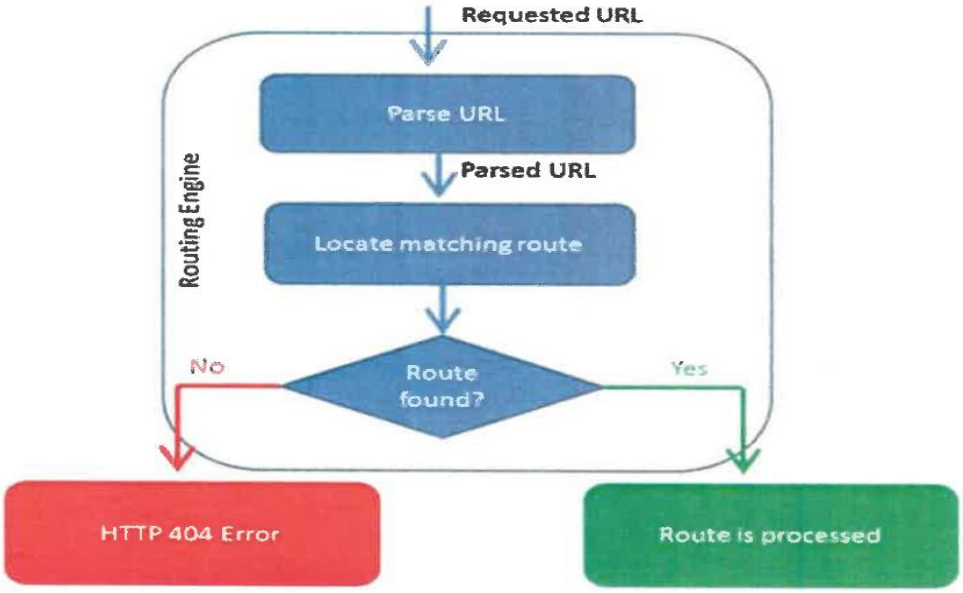

Figure 2. Functional Diagram of the Router component
[https://www.c-sharpcorner.com/UploadFile/3d39b4/routing-in-mvc/]

4. Implementation details

Currently, the web version of ExportValidator empowers the full grid of tests related to AMEF communication (the offline version). These tests involve extracting a digital certificate from a file exported by the fiscal machine, validating the certificate to the Order of the President of NAFA no. 146/18.01.2018 (certificate version, validity, structure), verifying the digital signature applied on the reported content, detaching the XML payload of the signed message, checking the

compliance of the XML file against the defined XSD schemes and comparing the fiscal data issued by the cash register with values predefined by the designated technician.

The program is built to allow three types of access: anonymous user (unauthenticated), authenticated user (technician), administrator user. In anonymous mode, the application only allows temporary operations, which are not stored in the database and are not associated with a test technician. This functionality is required to perform quick tests whenever the user only wants to check the structure of a certificate or the validity of a signature.

In the future, we intend to offer manufacturers of electronic fiscal equipment the possibility to access platforms in order to run conformity tests prior to submitting the equipment to ICI Bucharest for certification tests.

The authenticated regime represents the way in which compliance tests are run by ICI Buchrest technicians. The user has his own account to log in to the platform, where the following features are available:

- functions related to the extraction / validation of the digital certificate and signature;
- automatic testing of values reported by the cash register / fiscal equipment;
- view test history;
- account password renewal.

The last mode of the application is administration. It allows the administrator to register technicians and to establish permissions for each of them. Also, users logged in this regime, have access to all databases, can query all the tests performed and can modify or delete entries.

According to the operational procedure, the manufacturer / dealer that wishes to submit a fiscal electronic cash register / equipment in order to obtain the technical certification must attach the digital certificate corresponding to the equipment. However, in case it is not submitted to the test file, the ExportValidator application allows extracting it from a file exported by the fiscal equipment. Through an endpoint the user uploads a signed file according to the PKCS#7 standard. The uploaded file is saved on the server and a certificate extraction function is applied to it, receiving the location of the uploaded file and returning the digital certificate included in the analysed file.

For verification of the applied digital signature and the file structure, the fiscal electronic equipment communicates with the national computer system for surveillance and monitoring of the fiscal data by transmitting files in XML format. XML files submitted offline will be signed, using a digital signature included, compliant with the PKCS#7 standard, using the SHA-256 computational algorithm and the RSA 2048 signature algorithm. After the verification is completed, the payload (XML that contains tax data) is extracted from the signed message and displayed to the technician. For easier reading, it is possible to open it in a formatted way.

In case the electronic fiscal equipment is offline and cannot transmit data to the system, one or several files will be stored on an external storage medium that will be submitted by mail to NAFA. The existing file types are as follows:

1. reporting period file (A4200) – is a log file that declares the beginning and end of the reporting period;
2. daily report file for common machines (A4203) – represents the file containing the tax data reported by the fiscal equipment, others than those intended for taxi, exchange offices or airport shops that allow payment in foreign currencies;
3. daily report file for fiscal equipment intended for taxi – represents the file that contains the reported tax data for the taxi designated fiscal equipment;
4. daily report file for fiscal equipment intended for exchange offices – represents the file that contains the reported tax data for fiscal equipment designated to be used in exchange offices;

5. daily report file for fiscal equipment used inside airports, in shops that allow payment in foreign currencies – represents the file that contains the reported tax data for fiscal equipment used in airport shops that allow payment in foreign currencies.

The files must comply with the structure imposed by the Order of the NAFA President no.146/18.01.2018 and must comply with the sequence of elements. To establish the structure of the XML files, XSD schemes published on the website of the National Agency for Fiscal Administration were used. They aim to define the elements that can appear in a document, define the attributes that the elements in a document can have and fix the types of elements and attributes.

Determining the correctness of the XML file structure is done using schema defined as XSD. They are published by the National Agency for Fiscal Administration and must be complied with by all fiscal electronic cash registers/equipment in order obtain the technical certification. The test is performed within the same interface and at the same time as the validation of the digital signature.

In order to validate automatically the values, the test containing the exact sales registered by the device must be pre-configured. Internally, this is modelled using the Test Model in conjunction with its associated controller. This allows the use the Entity Framework, developed by Microsoft, to perform type operations Create, Read, Update, Delete on the application database. The test model contains all the properties of a report as well as a link to the associated file stored on the server. The fiscal identifier introduced in the pre-configured test must respect its property - 9 digits + 1 control digit. In order to avoid abnormal situations at validation time, we impose the constraint using the StringLength attribute (10). If the field does not comply with the conditions applied when submitting the form, the application will issue an error message.

The ExportValidator application uses a relational Microsoft SQL database for persistent storage of authentication data, existing roles within the application, and all tests run by technicians. To secure passwords, they were hashed using existing implementations in the UserManager class. A description of the application database is illustrated in the following Entity/Relationship diagram (Figure 3).

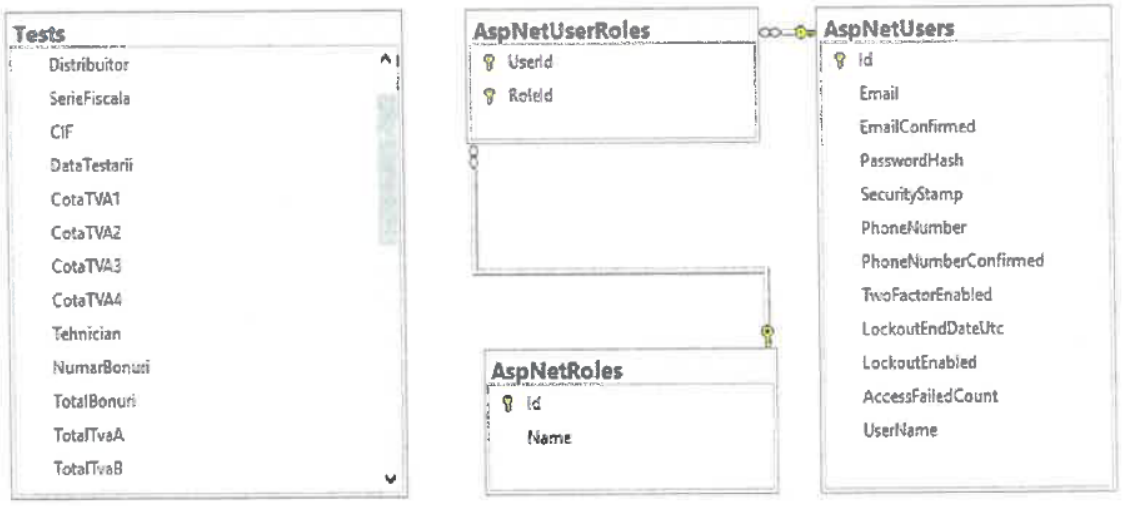

Figure 3. Database Entity/Relationship diagram

Next, two of the application's use-cases will be presented for a better example of its functionalities and its workflow (Figure 4).

Extracting the digital certificate

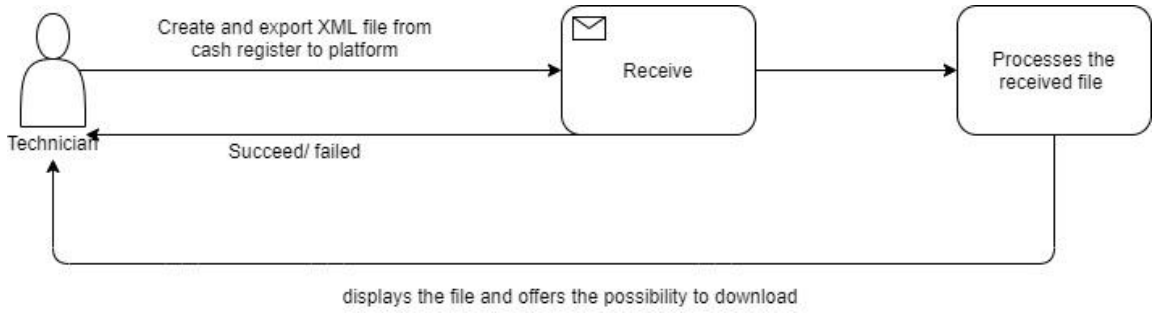

Figure 4. Workflow for extracting the digital certificate

In this scenario, the technician wants to extract the digital certificate of the fiscal equipment. The technician creates, exports and then uploads the XML file which contains the certificate from the cash register to the platform. The application receives the request, processes the XML signed file, and displays the results with the option to preview and download the certificate.

Validating the values reported by the fiscal equipment

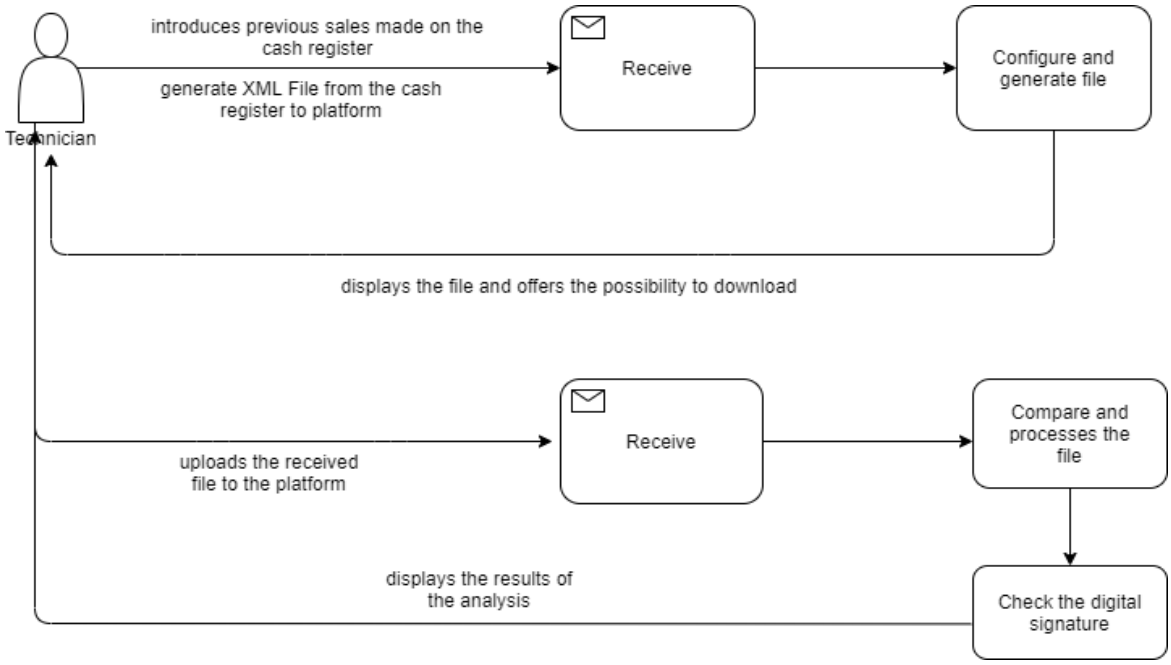

Figure 5. Workflow of analysing the sales registered by the fiscal equipment

Checking the sales on the fiscal equipment is done by comparing the XML file generated by it with the date configured by the tester (Figure 5). The electronic signature is also validated for the file generated by the equipment. After all these procedures have been performed, the platform displays the results.

5. Conclusions and statistic data

Based on data reported the regional structures of the National Agency for Fiscal Administration, there were approximately 1,500,000 cash registers in the records before 30.04.2018. This number includes both cash registers approved on the old architecture, as well as those approved using the ExportValidator application. The situation of electronic fiscal cash registers / equipment installed and taxed until 21.03.2019 is described as follows: a number of 322,465 were installed, which represents a 21% compared to the estimated 1,500,000. At the moment, the dynamics reflect a number of about 8,559 machines installed each week. Using this indicator, we can estimate the completion of the process of changing the fiscal electronic cash registers between April 2020 and

June 2022, depending on the actual number of cash registers requiring replacement. According to statistics provided by the National Centre for Financial Information, between September 10, 2018 and March 21, 2019, a number of 1,003,177 reports were received automatically from certified fiscal equipment. From this number, only 42,677 were classified as incorrect and rejected, resulting in a percentage of 4.25% of the total processed.

Implementation of the ExportValidator application within ICI Bucharest offered technicians responsible with testing and certifying electronic fiscal equipment an intuitive platform, using elements of the ASP.NET MVC web framework. This facilitated the verification of cryptographic mechanisms, by adapting the BouncyCastle library in the context of cash registers. Thus, it has significantly contributed to reducing the time and effort required in the process of certification of electronic fiscal cash registers / equipment.

REFERENCES

1. Albahari, J., & Albahari, B. (2012). *C# 5.0 in a Nutshell, Fifth Edition.* Sevastopol: O'Reilly Media, Inc.
2. Barbu, D. C. (2016*). Standards and Regulations Regarding Electronic Fiscal Marking Devices*. Romanian Journal of Information Technology and Automatic Control (Revista Română de Informatică și Automatică – RRIA).
3. Burd, B. (2012). *Beginning Programming with Java for dummies.* Canada: John Wiley and Sons.
4. Castle, T. L. (2016, November 11). *C# .NET Fips Resources*. Retrieved from www.bouncycastle.org: https://www.bouncycastle.org/fips-csharp/.
5. Groza, B. I. (2012). *Introduction to cryptography: cryptographic functions, mathematical and computational fundamentals.* Politehnica Timișoara.
6. Joydip, K. (2009, October 30). *Understanding the ASP.NET MVC Framework*. Retrieved from www.itprotoday.com: https://www.itprotoday.com/web-application-management/understanding-aspnet-mvc-framework.
7. Liberty, J., & Hurwitz, D. (2003). *Programming ASP.NET, Second Edition.* Sebastopol, California: O'Reilly & Associates.
8. Panda, M., & Nag, A. (2015). *Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux*. 2015 Second International Conference on Advances in Computing and Communication Engineering, 541-548.
9. Zamfiroiu, A., Cirnu, C.-E., Boncea, R., Rotuna, C., & Anghel, M. (2015). *Principles for the Design, Security and Administration of Cloud Storage Solutions*. Romanian Journal of Information Technology and Automatic Control (Revista Română de Informatică și Automatică – RRIA).

Paul-Cristian VASILE este absolvent al Facultății de Matematică și Informatică din cadrul Universității București, în prezent fiind student în ultimul an la programul de master Inginerie Software din cadrul aceleiași facultăți. Acesta deține funcția de Inginer Software la Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București. Experiența sa include domenii precum tehnologiile blockchain – unde a dezvoltat un instrument de cyber-diplomacy, platforme de e-Learning, metode de forensic digital și metode criptografice aplicate în domeniul caselor de marcat. Începând cu iulie 2020, este recunoscut ca "Certified Ethical Hacker", certificat emis de EC-Council.

Paul-Cristian VASILE graduated the Faculty of Mathematics and Informatics from the University of Bucharest and is currently a studying for the Software Engineering Master's degree. He holds a software engineer position at the National Institute for Research and Development in Informatics - ICI Bucharest. His work experience includes areas like blockchain technologies - where he developed a cyber-diplomacy tool, e-Learning platforms, digital forensics methods and cash registers cryptography. As of July 2020, he holds the Certified Ethical Hacker certification issued by EC-Council.