Promoting Quality e-Government Solutions by Applying a Comprehensive Information Assurance Model: Use Cases for Digital Signature

Ricardo Villalón-Fonseca[1(✉)], Alejandro Mora-Castro[1], Rodrigo Bartels-González[1], Miguel Carballo-Chavarría[2], and Gabriela Marín-Raventós[1]

[1] CITIC - ECCI, University of Costa Rica, San José, Costa Rica
ricardo.villalon@ucr.ac.cr
[2] Central Bank of Costa Rica, San José, Costa Rica

Abstract. Information and Communication Technologies, and specifically e-Government developments, occupy a relevant position in the national agenda of many countries worldwide. But development of such projects requires a careful planning, mostly when legal implications are bound to technological systems. This is the case when considering digital signature technology, as an essential element to support trust for e-government services. In this paper, we propose a comprehensive, systemic, and systematic, information assurance process to audit technological solutions for e-government services, and we show examples using digital signature from Costa Rica. Our approach supports applications developed with heterogeneous infrastructure and technologies. It also comprises the entire assurance process, including identification of threats and vulnerabilities, risk analysis, policy definition, and definition of security controls.

Keywords: e-Government · Digital signature · Information assurance · Private key infrastructure

1 Overview

Information and communication technologies (ICT) have been a priority development for many countries during last decades [1]. ICT enable benefits such as economic growth, sustainable development, government transparency, better life quality, and evolution of the information society [1]. Countries are developing national strategies for ICT that altogether are called e-Goverment, also known as e-gov, to deliver efficient and effective services to citizens [2].

In Costa Rica, a country of Central America, the Government started an e-gov plan, by 1998, to achieve the established economic, social, and educational goals [3]. In 2015, United Nations [1] recognized Costa Rica's efforts by awarding

F.J. Mata and A. Pont (Eds.): WITFOR 2016, IFIP AICT 481, pp. 223–234, 2016.
DOI: 10.1007/978-3-319-44447-5_21

the highest worldwide rise of 23 positions on the ICT Development Index (IDI), moving from global position 80 to 57. Although this is a good position for Costa Rica at the Americas, being the seventh of the region, there is still too much work to be done.

Sometimes there are doubts about the contribution of some ICT developments. For example, from our interviews with ICT government experts, we know that a poor or unrealistic design, or a lack of an appropriate project management, can generate issues such as defficient interoperability between different technologies, inappropriate infrastructure installed at the target context, insufficient system robustness, or security problems.

At the national level, some e-gov services require high quality implementations, because they are susceptible to legal implications. Failure to comply them can lead to lack of confidence from citizens, and ICT development may slow down, as a consequence. An example is digital signature, a mathematical scheme based on Private Key Infrastructure (PKI) [4], that is used to demonstrate the authenticity of a digital message or document. Digital signature is an esencial component of an e-gov infrastructure, but it relies on technology for a correct implementation.

Costa Rica already has a framework to sign legally binding documents using digital signature, through law 8454 [5] created on 2005. To support this law, Costa Rican government emitted the directive 067-MICITT-H-MEIC [6], by April 2014, to encourage and massify digital signature usage around the country. This directive asks the government institutions to provide online services through digital signature, as an alternative to handwritten signature. The directive gives a time frame of three years for the upgrade, but there is no regulation for quality or security assurance of existing or newly created digital signature solutions.

Digital signature solutions can be created with several different development paradigms, heterogeneous technologies, or different infrastructure platforms. This raises the concern of creating a national standard, with technical guidelines to evaluate the quality and assurance of these applications, given the legal implications, and the lack of technical regulations. Without some basic regulation, there is a risk of eventual trust issues in e-gov systems, that could jeopardize the development of national ICT strategies.

In this paper, we present the results of a research developed at the University of Costa Rica (UCR), in collaboration with the Costa Rican Ministry for Science, Technology and Telecommunications (MICITT), and the Central Bank of Costa Rica (BCCR) to create technical guidelines for improving the information security level of software applications having digital signature components, at government or private institutions. We propose a new comprehensive, systemic, and systematic, approach to analyze relevant scenarios of digital signature usage, in order to audit software applications created with heterogeneous software and hardware components. The proposal enables the entire assurance process, including identification of threats and vulnerabilities, risk analysis, policy definition, and definition of security controls.

We start with a literature review of relevant digital signature approaches at other countries. Then, we present the digital signature use cases and the application architecture examples used in the paper. After that, we explain our methodology for auditing or evaluating applications, followed by the results of applying the systemic and systematic assurance process, to show the advantages of our approach. Finally, we present the conclusions and future work to contextualize the results in a national scenario.

2 Literature Review

We did a review of several international solutions related to improving information security of software applications with digital signature components. Even though there are many countries implementing PKI solutions, we focused on those countries developing information assurance regulations, and technical guides for software applications in contexts requiring digital signature. At the end of this section, we summarize the contributions of the reviewed research.

In Brazil, information assurance of digital signature software is defined by technical manuals [7,8], which specify the certification requirements to be met. The requirements apply to software components for user authentication with digital certificates, as well as the creation and verification of digital signatures. Compliance with the security requirements is subject to a certification process [9], based on predefined tests [10,11].

The approach in France is to develop two certifiable protection profiles, based on the Common Criteria framework [12]. Profiles are defined for signature creation applications [13], and signature verification modules [14]. They aim to protect critical resources, such as the electronic document, and all its intermediate representations, the digital certificate, the digital signature, and the signature attributes. Conformity with protection profiles can be certified [15].

Spain also uses Common Criteria protection profiles to asses and certify digital signature applications. There are four protection profiles [16–19], which define security requirements for digital signature creation and verification software that uses the National Electronic Identity Document (DNI-e) as the secure signature device. These profiles focus on the protection of different assets, such as the electronic document and its intermediate representations, signature attributes, signature verification data, signature policies, and authentication data to use the DNI-e. Concordance with protection profiles is certifiable [20].

In contrast, Belgium uses an open source approach to audit its digital signature applications. Belgian citizens obtain, at the age of twelve, an electronic identity card, known as eID, which contains digital certificates [21]. Federal Public Service for Information and Communication Technology (Fedict) is an organization that develops different ways to promote usage of eIDs. They selected an open source approach, and collaborate intensively with software developers to create digital signature applications. The source code of eID initiatives is available through guidelines, and can be accessed by interested parties, which can propose and make improvements [22].

In summary, we found operational solutions whose objectives are comparable to ours. We identified relevant information security aspects that must be protected as part of the information assurance process, such as handling of digital certificates, integrity of the data (signed and to be signed), and confidentiality of the credentials required to access the encryption devices. Therefore, the creation of a national regulation, and technical guidelines to evaluate the information assurance process of digital signature applications seems to be a must-have.

3 Digital Signature Scenarios

From the literature review, and additional considerations within the Costa Rican context [6], we selected the digital signature use cases to be analyzed. In this section, we briefly describe these scenarios, and we use one of them to explain how our methodology works.

3.1 Selected Use Cases

We are interested on four digital signature use cases because of their relevance in the context of software applications. The scenarios are:

1. *Digital signature creation*: creation of a set of electronic data to be attached to an electronic document, in order to identify the signer unequivocally, and ensure the integrity of the signed document.
2. *Digital signature verification*: validation of the identity of the signer, the integrity of the signed document, and the validity of the digital certificate used to create the signature.
3. *User authentication with digital certificates*: process of demostrating ownership of a private key, with the purpose of validating the user identity.
4. *Conversion from basic to advanced format*: addition of attributes to a signed document, in order to ensure long-term verifiability of the contained digital signatures.

Following, we describe the details for a simplified version of the digital signature creation process, and we use it in next sections, as an example, to apply our methodology. The remaining use cases are solved using the same methodology.

Digital signature creation typically starts when a user picks an electronic document to be signed, and his or her digital certificate for identification. The certificate is validated using a set of predefined criteria. Before signing, a hash function is applied to the document to generate a digest. Then, the digest is encrypted using the private key of the user, which produces the digital signature. Finally, the original document, the digital signature, and the digital certificate are put together to create a signed data object, also known as the signed document, as shown in Fig. 1.

Digital signature creation use case shown in Fig. 1 can be implemented in several diferent ways, by using heterogeneous technologies, software architectures,

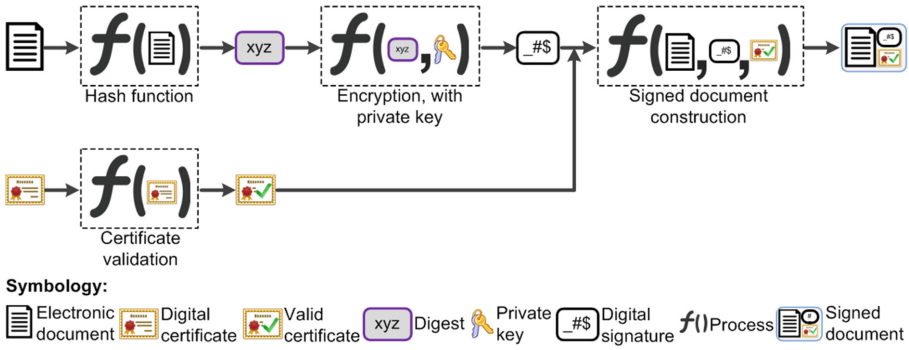

Fig. 1. Digital signature creation process.

or different infrastructures. In next section, we show two example implementations for this use case. Even though both implementations follow the same process, they may cast diferent risk analysis. This shows that, at a large scale, it could be hard to audit a large number of applications having very different implementations. We will use the example to show how to deal with this problem by using our methodology.

3.2 Application Architectures

There are multiple variables influencing the design of a software application to digitally sign a document. The development paradigm, the programming languange, or even the infrastructure platform are examples of these variables. The relationship between the software application and its possible implementations is one-to-many. For example, consider two different architectures to implement a digital signature creation application, as shown in Fig. 2.

The standalone architecture, Fig. 2a, represents an all-in-one application, installed in the computer of the user, and all the required software components to create the digital signature are located in the same computer. In this example, the secure cryptographic device, which stores the private key of the user, is connected to the computer using a USB cable.

On the other side, the client-server architecture, in Fig. 2b, represents an application with distributed software components across different machines. Some actions take place on the user side, but other functions are executed on the application server, and they are remote from each other. In this case, the secure cryptographic device is also remotely accessed, by using a network communication.

Both architectures implement the same use case for digital signature. But they are exposed to different information security risks, because of the vulnerabilities or threats that can show up at the different components of each architecture, and because they also have different communication channels between components. For example, asking the cryptographic device to sign a digest requires a

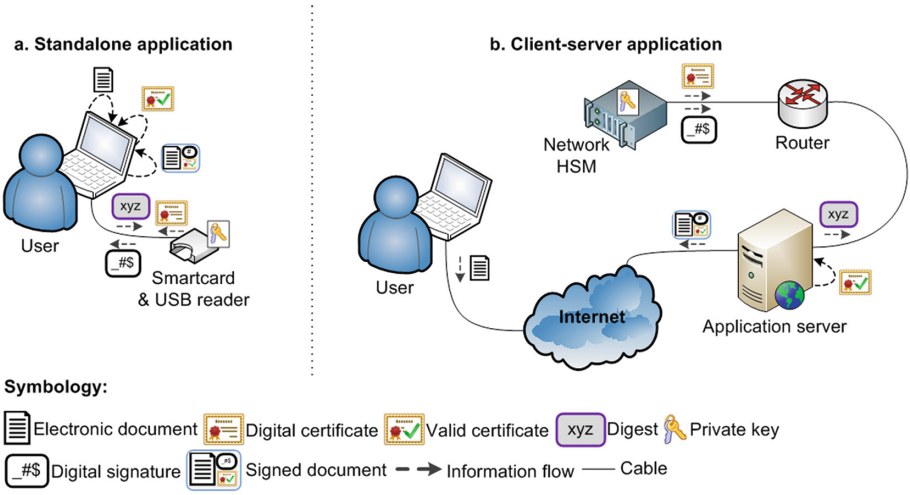

Fig. 2. Standalone and client-server architectures implementing a digital signature creation process.

local communication for the standalone application, but it is a remote operation for the client-server application.

In the next section, we describe a methodology to deal with these differences in a systemic and systematic way. We propose an assurance solution for the entire system, using a systematic step-by-step process to deal with the different risk analysis inherent to each architecture.

4 Methodology: InfoSec-Tree Model with Extensions

We selected the InfoSec-Tree Model [23] as the main tool to establish our information assurance process. We extended the original model, and improved the systemic capabilities provided by its whole-parts structure. We propose new ideas to use the model in a wider context, as a tool for risk analysis. We also add new elements to support the systematic auditing of a large number of software applications, with similar security requirements, but having different software components, implementation techniques, and using different software development tools.

4.1 The Basic Model

The InfoSec-Tree Model is an information security model proposing a hierarchical whole-parts approach. In this model, a software application or technological system is decomposed into its parts, using a tree structure. The root node represents the entire system being assured. Child nodes are created by decomposing

the root node into its components, and each component can be divided into subcomponents, and so on. The depth of the tree represents the required assurance granularity level for the system or its subcomponents. The only restriction of the hierarchical structure is to keep consistent the whole-parts relation between each node and its children.

The InfoSec-Tree Model proposes a security triad abstraction to specify security requirements inside each node of the tree, or between pairs of communicating nodes. Triads representing security requirements for information contained inside a node are called internal-triads. End-point triads represent security requirements of information being transmitted between two nodes. Triads have a structure like *(assurance moment, information state, security service)*, to represent the time dimension of the event, the state of information, and the security requirement, respectively.

Information flow diagrams are another important component of the InfoSec-Tree Model. They represent communicating nodes, connected through endpoint-triads. Information flows describe scenarios of information transmitted between pairs of communicating nodes, and also provide the elements to define information security controls in this context.

There are some interesting facts about the InfoSec-Tree Model that permit us think on it as a solution:

- The whole-parts structure enables a systemic security analysis of a software application. The nodes can represent modules or components of the application, at any level. This feature facilitates a security analysis of several different software applications using a consistent methodology.
- The target system can be composed of hardware, software, or a mix of them, and it also can be of any size. This enables an homogeneous way to define an assurance process comprising a software application, and its technological infrastructure.
- The original model is designed to define security controls but, in our experience, the underlying structure can be extended, in a natural way, for risk analysis, threats and vulnerability identification, security policy definition, and other additional components of an information assurance process.

Following, we explain the main ideas for the proposed extensions to the InfoSec-Tree Model.

4.2 Adding Vulnerabilities, Threats and Risk Analysis

The original InfoSec-Tree Model proposes a solution to define information security controls for an entire computer system. But, after using the model for a while, we found that it can also be used to precisely set the location of identified vulnerabilities and threats in a system. Consequently, the tree structure is a good place to locate and identify security risks. Triads represent places where information is located, so we can use them to identify risks, and we can pinpoint their location accurately, by indicating the specific triads where vulnerabilities and threats occur.

4.3 Defining Control Objectives

The result of applying the original InfoSec-Tree Model is a set of specific security controls, located at different points, of a specific application. But we require a wider definition for the concept of security control, because we need to audit the security of a large number of applications, having different architectures and infrastructures. We need to establish a minimum but appropriate set of security requirements for many software applications having digital signature components. Then, we broaden the security control concept to control objective.

A control objective is a generic definition of a security control to specify a security countermeasure. It excludes specific implementation details such as used technologies, algorithms, implementation languages, or other specific technical details. Details are only included when they represent a relevant aspect to reach the goal indicated by a security requirement. This new definition provides an appropriate level of generality to define security controls at a large scale.

Additionally, control objectives enable usage of tools that generally apply, on industry, during an information assurance process. For example, in-depth or layered security [24] can be defined in a natural way at consecutive nodes of an infoSec-Tree. Other strategies, such as principle of least privilege or least common mechanism [24], can be also be applied. Therefore, we can establish generic but consistent security requirements, over a large number of different applications.

4.4 Analyzing Risks in Information Flows

While doing risk analysis for information flows, with the original InfoSec-Tree model, we obtained different risk assessments for the different application architectures. We identified systematic differences for applications having local (centralized) or remote (distributed) communicating components. For example, our standalone application has only local communications between components, for the digitial signature creation use case. But the client-server application has some remote communications between nodes, like the communication between the server and the cryptographic device, and consequently risk assessment is different.

Then, we propose to differentiate risk analysis base on local or remote communications. We can have the same information flow diagram, but different risk analysis and security controls for different architectures. This way, we keep a systematic assurance process, based on how local or remote are the flows, but at the same time enabling analysis of different application implementations. Consequently, we can generate appropriate technical guidelines for an assurance process using a standarized methodology.

5 Results

In this section, we show the results of testing our methodology, for the digital signature creation use case, with a standalone architecture and a client-server

architecture. First, we describe the infosec-tree and information flow diagrams. After that, we explain how the systemic and systematic approach arises. Finally, we show the risk analysis process, and compare the results between our example architectures.

The infosec-tree for the standalone architecture from Fig. 2 is shown in Fig. 3a. The root node Digital Signal Creation System represents the entire system we want to assure. It comprises the end-user Computer, the Secure Cryptographic Device, and other components that we are not interested, represented as the Rest of the System. The Digital Signature Application is part of the Computer, and it is composed of the Signature Creation Components, and other elements (Rest of the App). The Signature Creation Components is composed of seven smaller elements that we are going to use when creating a digital signature.

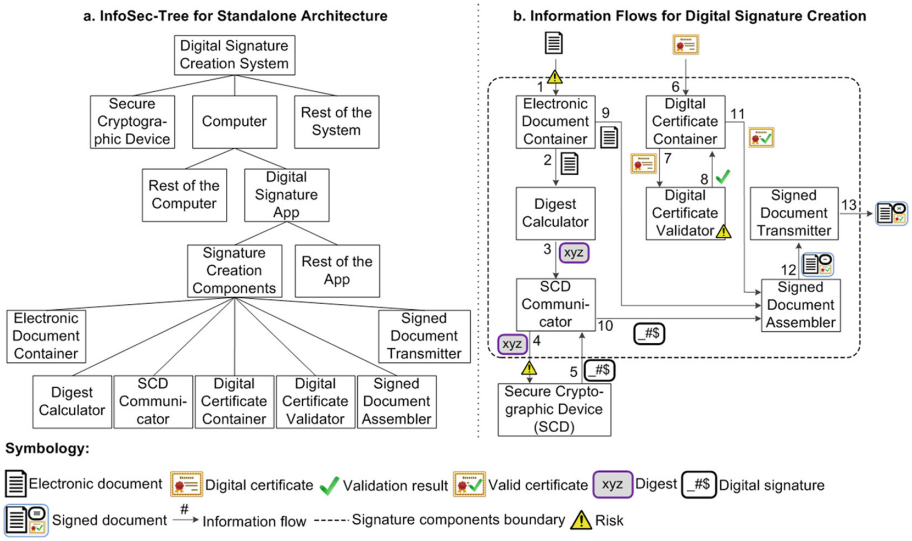

Fig. 3. Infosec-tree and information flow diagram for the digital signature creation process.

The infosec-tree for the client-server architecture is not shown, but it has a corresponding tree structure, with a user-side and a server side, instead of having all components at the user-side. In that case, the Secure Cryptographic Device and the Digital Signature Application with its components are located at the server-side. The user-side contains the end-user Computer, a Web Browser, and other elements to appropriately communicate with the server.

Even though the two implementations have different architectures, and consequently different infosec-trees, they have the same information flow diagram, shown in Fig. 3b, for creation digital signature use case described in Fig. 1.

The flow starts when an electronic document is provided (1 in Fig. 3b), and is temporarily stored in an Electronic Document Container. There, the document

is sent (2) to a Digest Calculator, to produce a digest with a hash function. The digest is passed (3) to a Secure Cryptographic Device Communicator (SCDC), which transmits (4) the digest to a Secure Cryptographic Device (SCD). The SCD encrypts the digest with the private key of the user, and produces a digital signature that is sent back (5) to the SCDC. On the other hand, the application also receives (6) the digital certificate of the user, which is temporarily stored in a Digital Certificate Container, and validated (7, 8) by the Digital Certificate Validator. Then, the original document, the digital signature and the digital certificate are put together (9, 10, 11) using a Signed Document Assembler to generate the signed document. Finally, the Signed Document is sent (12) to its final destination (13) with a Signed Document Transmitter.

The infosec-tree enables a systemic assurance process, because the whole-parts structure naturally covers the entire system. From a security perspective, a systemic approach helps improve confidence levels in the system. Furtheremore, the information flow diagram promotes a systematic assurance process, because we can interate over all nodes of the diagram, and over all connections between nodes, doing an assurance process for each corresponding component or communication channel. Internal triads of the original InfoSec-Tree Model can be used to specify security aspects for a component, and they can be extended to include information about the associated vulnerabilities, threats, and risks, as described in our methodology. In the same way, end-point triads can be used to specify security aspects for the communication channels.

After iterating over the components and communicating links of the system, we obtain a detailed list with our security concerns, such as vulnerabilities, threats, and risks, that we can locate precisely, at the corresponding elements of the flow diagrams. Then, we can move on to consider security policies, and use them to define the appropriate security control objectives. As an example, in Fig. 3b, we show three identified risks, using warning symbols, for information flows 1 and 4, and at the Digital Certificate Validator node.

The risk for information flow 1 identifies eventual modifications (integrity) to the electronic document to be signed, when the document is going from its source to the Electronic Document Container. The risk for information flow 4 identifies eventual modifications (integrity) to the digest, while moving towards the cryptographic device for signing. The risk at the Digital Certificate Validator identifies eventual problems when validating the digital certificate, due to an incorrect implementation of the software component.

Now, we can compare how the security issues associated to the identified risks can impact the information security of the system. The application architecture is a relevant aspect at this point. For example, an attacker trying to intrude information in flows 1 or 4, with a standalone implementation, will have to intrude local communications between the CPU and a local device, such as the hard disk or a USB cryptographic device. In contrast, for the client-server architecture, the flows are network communications that can be intercepted with a man-in-the-middle attack, unless we encrypt the communication channels appropriately. The risk identified at the Digital Certificate Validator has a similar impact for both

architectures, because interactions to validate the certificate are very similar in both implementations. After assessing the impact associated with the risks, we can proceed to define the corresponding control objectives.

6 Conclusions and Future Work

We found, from the literature review, that our initiative to create a national regulation, with technical guides for e-gov solutions, is accepted and well supported in developed countries having important e-gov strategies. We tested our methodology with use cases in the context of digital signature applications. We determined risk scenarios for information security, in a simplified way that can be applied to groups of heterogeneous software applications but having similar goals. After identifying risks, we evaluate impact and define appropriate control objectives.

Our methodology is comprehensive and systemic, it can be applied in width and height, to all components of the systems, and up to the desired level of granularity or detail. But it is also systematic, by its organized way to analyze the security requirements. The systemic and systematic behaviours enable us to define well supported technical guidelines, with wide coverage and suitable confidence levels for large-scale scenarios.

In summary, we consider that our information assurance methodology is appropriate to develop technical guides supporting regulations at a national scale. This way we can improve the quality and security of ICT projects, such as developing ICT national strategies with e-government services.

For future work, we are working on a first draft of the technical guidelines for digital signature applications, and we expect to use the methdology in other information security contexts. Finally, we are proposing the creation of software tools that help to automate usage of the assurance model, so we can improve auditing processes, and also adapt better for new emerging architectures, and technological platforms.

References

1. International Telecommunication Union: Measuring the Information Society Report. ITU, Geneva (2015)
2. United Nations: United Nations E-Government Survey 2014 - E-Government for the Future We Want. United Nations, New York (2014)
3. Alvarado, L., Garro, A.: Gobierno electrónico en Costa Rica. Tesis para optar por el grado de Licenciatura en Derecho. Facultad de Derecho, pp. 252–260. Universidad de Costa Rica (2009)
4. Buchmann, J., Karatsiolis, E., Wiesmaier, A.: Introduction to Public Key Infrastructures. Springer, New York (2013)
5. Gobierno de Costa Rica: Ley de Certificados. Firmas Digitales y Documentos Electrónicos. Diario Oficial La Gaceta. Imprenta Nacional, San José (2005)
6. Gobierno de Costa Rica: Masificación de la implementación y el uso de la firma digital en el sector público costarricense. Diario Oficial La Gaceta. Imprenta Nacional, San José (2014)

7. Instituto Nacional de Tecnologia da Informação: Manual de Condutas Técnicas 4 - Volume 1: Requisitos, Materiais e Documentos Técnicos para Homologação de Softwares de Assinatura Digital no Âmbito da ICP-Brasil. 2.0, São Paulo (2007)
8. Instituto Nacional de Tecnologia da Informação: Manual de Condutas Técnicas 5 - Volume I: Requisitos, Materiais e Documentos Técnicos para Homologação de Softwares de Autenticação no Âmbito da ICP-Brasil. 2.0, São Paulo (2007)
9. Instituto Nacional de Tecnologia da Informação: Homologações. http://www.iti.gov.br/servicos/homologacoes
10. Instituto Nacional de Tecnologia da Informação: Manual de Condutas Técnicas 4 - Volume II: Procedimentos de Ensaios para Avaliação de Conformidade aos Requisitos Técnicos de Softwares de Assinatura Digital no Âmbito da ICP-Brasil. 2.0, São Paulo (2007)
11. Instituto Nacional de Tecnologia da Informação: Manual de Condutas Técnicas 5 - Volume I: Procedimentos de Ensaios para Avaliação de Conformidade aos Requisitos Técnicos de Softwares de Autenticação no Âmbito da ICP-Brasil. 2.0, São Paulo (2007)
12. Common Criteria: About the Common Criteria. https://www.commoncriteriaportal.org/ccra/
13. Agence Nationale de la Sécurité des Systèmes d'Information: Protection Profile Electronic Signature Creation Application, 1.7 (2011)
14. Agence Nationale de la Sécurité des Systèmes d'Information: Protection Profile Electronic Signature Verification Module, 1.7 (2011)
15. Agence Nationale de la Sécurité des Systèmes d'Information: Certification Critères Communs. http://www.ssi.gouv.fr/administration/produits-certifies/cc/
16. INTECO: PPSCVA-T1, EAL1. Perfil de Protección para la aplicación de creación y verificación de firma electrónica Tipo 1, con control exclusivo de los interfaces con el firmante y nivel de evaluación de los requisitos de seguridad EAL1. 2.0 (2009)
17. INTECO: PPSCVA-T1, EAL3. Perfil de Protección para la aplicación de creación y verificación de firma electrónica Tipo 1, con control exclusivo de los interfaces con el firmante y nivel de evaluación de los requisitos de seguridad EAL3. 2.0 (2009)
18. INTECO: PPSCVA-T2, EAL1. Perfil de Protección para la aplicación de creación y verificación de firma electrónica Tipo 2, con nivel de evaluación de los requisitos de seguridad EAL1. 2.0 (2009)
19. INTECO: PPSCVA-T2, EAL3. Perfil de Protección para la aplicación de creación y verificación de firma electrónica Tipo 2, con nivel de evaluación de los requisitos de seguridad EAL3. 2.0 (2009)
20. Certification Body: Functional Certification. https://oc.ccn.cni.es/index.php?option=com_content&view=article&id=52&Itemid=55
21. Fedict: The eID. http://eid.belgium.be/en/find_out_more_about_the_eid/the_electronic_identity_documents/the_eid
22. Fedict: The Approach. http://eid.belgium.be/en/developing_eid_applications/the_approach
23. Villalón-Fonseca, R., Solano-Rojas, B., Marín-Raventós, G.: Infosec-Tree Model: an applied, in-depth, and structured information security model for computer and network systems. J. Internet Technol. Secur. Trans. (JITST) **3**, 300–310 (2014). Infonomics Society, Londres
24. Bishop, M.: Computer Security: Art and Schience. Addison Wesley, Boston (2002)