Abstract: In this work, a solution to the problem: How to extend the current Cartão de Cidadão’s certificates infrastructure in order to associate a citizen’s electronic identity with the role he plays in society; is researched and several guidelines for role certification are discussed. The adopted solution was based on a government trusted entity that assists the current National Identification Provider with the aggregation of roles and attributes and offers role certification for digital signatures that have claimed (but not proven) roles. This work uses only open-source material and hopes to assists other countries/entities in the field of role certification.

Keywords: Cartão de Cidadão, Role Certification, STORK, SAML2, DSS, XAdES.

Introduction:

The main goal of the e-ID Cartão de Cidadão (CC) card is to offer the possibility of a citizen to be able to prove that he is the citizen he claims to be to another person or machine in an unsecure network, and once that goal had been achieve a problem or emergent necessity rouse immediately: How to prove to the other person or machine that the citizen is who he claims to be with its roles/attributes that he claims to have? This is something that is "still green" since the electronic identification (eID) technology is not mature itself and requires some research.

This thesis’ main problem is about this necessity and revolves on the question: How to extend the current CC’s certificates infrastructure in order to associate a citizen’s electronic identity with the role he plays in society, offering role/qualified web authentication and digital signing, and in the mean time assure that this association is not fraudulent?

In the process of developing a solution, several manners of solving this solution where studied and discussed so that this work could serve as a best practices guide for other countries or entities interested in this problem. The development of this solution is also meant to be used has a pilot project for the “Agência para a Modernização Administrativa – AMA” organization, and has other contributions in mind, such as: Establishing a best practices guide for roles certification; offer a study and analysis of open norms and technologies that can create a solution to the problem in a low cost and flexible way; expanding the Portuguese e-ID services and extend the Security Assertion Markup Language (SAML) by designing two new profiles.

Related Work:
In this article the related work will only mentioned the more relevant standards and in order to keep this article short and succinct, only the conclusions of this study will be presented.

**XML and CMS signatures:**

Comparing the two standards: They both serve the purpose of digitally signing any type of data using qualified certificates like the ones the CC has, both can be used in digital signature formats such as XML Advanced Electronic Signatures (XAdES) and CMS Advanced Electronic Signatures (CAdES) and they both allow the storage of arbitrary attributes such as signing time, etc., but the fact is that XML Signature is more broadly used since it can be used by almost every XML based standard and that makes it more flexible and likely to be supported by other parties than the Cryptographic Message Syntax (CMS) signature. This thesis makes use of SAML, Digital Signature Services (DSS) and other XML based standards which XML signature is also part of, the choice of XML signature over CMS is made not because one is better than the other, but because most of this thesis’ technologies use XML signature for signature purposes.

**XAdES, CAdES and PAdES:**

Between the 3 types of digital signature technology, Portable Document Format (PDF) Advanced Electronic Signatures (PAdES) is the most different of them since it only signs PDF documents, rather than any type of data. Most of the common documents formats aren’t prepared to include signature data; this is where CAdES and XAdES come into play, offering a standard method to keep together the sign data with the document data. PAdES on the other hand, allows this by extending the PDF format and adding the signature to the document itself, which means that it is not necessary to a second application, like CAdES or XAdES, to read/understand the sign data and retrieve the document data. Since the widespread usage of PDF on the internet and the mentioned features of PAdES (see section: 3.8.3 of the full article), this signature format is quite convenient, but another format is needed to be able to sign all kind of data; CAdES and XAdES are very similar to each other, although XAdES is actually more flexible (or can be easier modified) and simpler to implement. [1]

The Advanced Electronic Signatures (AdES) family allows a user to add qualification/role data to the signature itself, which means that the use of attribute certificates is not mandatory if upon the signature process, the solution can verify if the role(s) that the user is claiming to have, is(are) indeed true.

**DSS and EPM:**

With the use of a DSS combined with attribute certificates the management and distribution of these certificates can be substantially simplified. Though has seen in the previous section, the AdES format can be a substitute for these certificates and since DSS can support any type of the AdES family format, the combination of AdES with a DSS server (DSS implemented with AdES Profile) connected to the CC’s Public Key Infrastructure (PKI) and Attribute Providers (AP) turns out to be a simpler an easy solution.
Electronic PostMark (EPM) by the other hand can offer the same features as DSS, in terms of signature provider and can also provide with workflow logging, non-repudiation data storage, graphical image of the EPM signature/certification, etc. (see section 3.9.2 off the full article for more information), but it is a standard planed to solve the needs of Postmarks companies, its best application is to certify the authenticity of e-mails and its contents, not a simple case of a user who wants to sign its company report and share it (not necessarily by e-mail) later with his boss, that needs to verify the authenticity of the document.

Given this, DSS ends up offering a more suitable solution, since it can incorporate EPM capabilities through the implementation of EPM Profile and the use of Time-Stamping.

**SAML and OpenID:**

The use of SAML in this thesis could be questionable as if it could be replaced by another approach like OpenID\(^1\), this standard is designed specifically to enable Single Sign-On authentication and thus it’s simpler to implement (since it’s a concrete solution for a concrete problem) and by not relying on XML to exchange data between parties (it conveys a set of "key-value" pairs and sends them within HTTP messages) its lighter to parse then the SAML approach. Though it has its advantages SAML is a more abstract and flexible solution and in fact comparing SAML with OpenID is only fair when talking about a specific profile of SAML, the Web SSO profile, since it’s the specific profile that offers a solution to the same problem.

In short words, SAML offers an array of security features, it is more extensible and can express subtle, hierarchical relationships, something that is difficult to accomplish with simplistic key-value pairs. It might not be as simple to implement has OpenID but it offers a more daunting solution beyond enabling solely SSO and offers a better solution to this thesis problem since it can easily accommodate the necessary attributes to be exchanged between parties. [2]

Also, for this thesis’ requirement OpenID would not be the best choice, since OpenID is built for user authentication using front channel communication, it is dependent on the user’s interaction. This thesis’ solution is based on the safely exchange of roles/attributes between servers through back channel communication and thus the OpenID protocol would serve no purpose.

**Solution:**

This solution will be tested and validated within the “Agência para a Modernização Administrativa” (AMA), where it has been named: “sistema de CERTificação electrónica de Atributos” (certA). The system is based on an attribute aggregator entity that using the CC’s PKI and trusted Attribute Providers, offers the

---
\(^1\) [http://openid.net/](http://openid.net/)
capability to create digital signatures with qualification data counter-signed and verified by this trusted entity (certA).

Figure 1 - Solution’s Overview

Figure 1 shows, what entities are involved and how they interact with each other. It does not represent the detailed view of the flow of information.

With the purpose of offering role based authentication and digital signatures with attributes, two scenarios come into play: Web Role Authentication and Digital Signature Role Certification. Currently this is already possible, but with no means to have absolutely sure that the claimed roles/attributes are true. So a government certifying entity is needed and in this case it is called certA.

**Web Role Authentication:**

The existing technology that allows government certified user authentication is based on typical SAML Single Sign On (SSO) operation, involving a Service Provider (SP) and a government Identification Provider (IdP). In a SAML SSO scenario a SP can request for attribute(s) verification along with the authentication request, the problem is that the current governmental IdP has no means to verify all the existing attributes that a citizen can have, and does not know which entities (from all the existing entities) it must contact in order to verify the requested attributes.

What this solution proposes is to build an entity that does just that: knows all legal existing roles/attributes and knows all the entities responsible for actually verifying if the citizen has access to those attributes. So when the National IdP receives an authentication request that also requires attribute verification, the IdP (after the user's authentication), requests certA for the verification of those attributes reliving the IdP from that “burden”. Also, part of this proposal is that this entity becomes detach from the IdP (as shown in Figure 1) in order to offer this attribute certification service to other trusted entities.

**Digital Signature Role Certification:**

Through the CC certificates emitted by the “Entidade Certificadora do Cartão de Cidadão” (EC CC) and through governmental Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP) entities, it is possible to create a digital signature that bears the government credibility that the signature
truly belongs to the claimed citizen. Currently, signature document formats such as XAdES, can incorporate extra information, like roles, on the created signature, the problem is (like on the web authentication case) how to prove that those claimed roles are true? One can prove that the signature belongs to the citizen, through the use of the CC’s PKI, but there is not responsible entity that can verify is the user is actually (for instance) an architect, like the claimed role that the citizen used, says.

The proposal for this problem is to build a signing applet (or a plug-in or even a standalone application) that creates digitally signed documents with roles (chosen from a list of all the, acceptable by the government, roles), signed by the CC’s signature certificate and then sends the document to a government signature provider that counter-signs the signed document, after verifying the authenticity of the claimed roles, through the use of the attribute certification service mentioned above, and sends the counter-signed document to the user. With this new document, signed by the user’s CC certificate proving the identity of the user and with the certA’s certificate proving the authenticity of the roles/attributes claimed in the first signature, the citizen can now send the document to another person/entity which in turn can use the signing applet (or a plug-in or even a standalone application) to verify the document’s authenticity through the use of the verifying services of the government signature provider.

**Open technology chosen to implement the proposal:**

To implement back-channel communication from IdP to certA and from certA to AP’s, SAML will be used, and two new Profiles will be implemented (along with the SOAP over HTTP binding), to ensure security measures and status information for each attribute. Communication between entities will be made under HTTPS to assure data confidentiality.

For digital signature purposes, DSS will be used with the profiles: AdES and EPM implemented and HTTP binding for communication. Communication between the DSS server and the signing applet will be made under HTTPS too.

For creating digital documents the signing applet will be using the XAdES standard.

*Remarks:* This solution runs under the premise that the Attribute Providers are government or legal trusted entities, otherwise certA cannot trust them. The use of SAML has also the purpose to allow interoperability between certA and Pan-European Proxy Service (PEPS), a Secure idenTity acrOss boRders linkeD (STORK)’s component.

**Summary:**

In short terms, this solution predicts the use of a government certified entity that offer aggregation and verification capabilities of roles and attributes recognized by the government that a citizen can inherit. This entity can then be used by trusted IdPs and other entities like a Signature Provider that requires role certification capabilities. This Signature Provider is also a component of the solution and in combination with the first mentioned component, offers the capability to certify signed documents with claimed roles.
Implementation:

As mentioned before, this solution makes use of 3 open-source standards, SAML, DSS and XAdES. All 3 technologies are needed in different entities and therefore where made into java API’s: certA-SAML-API, certA-DSS-API and XAdES API; this last API is part of the certA-DSS-API.

The Figure 2, show the solution’s component diagram where the reader can see that with the exception of the signing applet, all the other components assume a server like behavior are since all of the programming was done in java, a glassfish container was used to house the web java application: certA.

Note: For testing and development purposes it was necessary to also implement an IdP and several APs and therefore they were also put on a glassfish container.

The java APIs are used by the following components (represented in Figure 2):

- IdP can use (if necessary) the certA-SAML-API in order to use the SAML Aggregation Attribute Profile to communicate with certA;
- Signing Applet uses the certA-DSS-API and certA-XAdES-API to sign documents and send them to the certA DSS component;
- certA uses all 3 APIs, to understand the IdP’s request, communicate with the trusted AP’s, understand the Signing Applet’s request and to be able to counter sign XAdES documents;
- AP can use (if necessary) the certA-SAML-API in order to use the SAML Safe Attribute Profile to understand the certA’s request.

New SAML Profiles:

Note: this is a small description of what the two new SAML profiles do and how they work is presented in this section. For a full understanding of these profiles, see the full specification in the sections 6 and 6.2 of the full article. Also for a better understanding, see sections 5.6.1 and 5.6.2 of the full article.

As seen in Figure 3, the Safe Attribute Profile works on a peer-to-peer level and the Attribute Aggregation Profile is concerned with the overall interaction between the 3 types of entities involved.
The Safe Attribute Profile rouse from the necessity of a SAML profile that assure safe exchange of data between entities; the SAML Core [3] (document that specifies the SAML Attribute Query/Response) does not require for Elements like Signature or Destination to be present in a Request or Response. Also, the requirement for a status code that reflects the attribute current state is necessary for most of the attribute transactions that two different domains might undertake. So due to since no SAML Profile specifies this, a new one was created. This profile is in its essence the Attribute Query Profile and like its "parent" is only concern with point-to-point attribute transactions, having no “memory” or “conscious” that the transaction made is part of a bigger agenda.

So another Profile was created: Attribute Aggregation Profile. This profile uses the Safe Attribute Profile in order to safely exchange attributes between each entity, and keeps track of what is happening between the IdP, certA and AP’s. This profiles prevents misusage from the IdP, by not allowing the request of verification of all the known attributes of a subject (the Attribute Query Profile allows this, and so does Safe Attribute Profile), and specifies a SAML response that gathers all the verifies and unverified attributes as well as their status code, reporting what went wrong with the unverified attributes.

CertA-SAML-API:

This API was made to help developers make their IdP’s and AP’s interact with certA, simply by calling one method instead of having to study the new SAML Profiles in order to make their solutions work with certA; otherwise this API would be made into a java web standalone application. It makes use of java OpenSAML\(^2\) API in other to fully implement the two new designed SAML profiles and is planned to be used by a web java application running in glassfish (certA).

XAdES API:

\(^2\) https://spaces.internet2.edu/display/OpenSAML/Home/
A XAdES API, made also in java, “helps” the signing applet to create XAdES documents with roles and add a time-stamp issued by a TimeStamp Authority (TSA), in one method call; and also “helps” the DSS server to create a counter signed XAdES document as well as verify the signed or/and counter signed document’s signature and validity of the certificates used on those signatures.

**CertA-DSS-API:**

The DSS java API processes basic DSS Requests/Response and implements the DSS Core [4] and DSS Profile AdES [5]. It was planned to be used by both certA and signing applet with the use of a single method call; it makes use of a XAdES API in order to sign, counter signed and to be able to verify the signatures and certificates’ validity.

The DSS API understands and knows all of the specified DSS Core and DSS Profile AdES Elements, but in fact does not supports all of them, though they are not actually needed since the DSS server is designed to do two specific things only: Verify the authenticity of counter sign documents created by the DSS server and counter signing XAdES documents, containing roles, signed by the citizen’s CC signature certificate; adding optional requests (or requests for an CMS signature) to the DSS server is meaningless since the server is designed to do only what the certA requires and not what the user/client wants.

**Case Study and Validation:**

Using a virtual machine running windows 2003 server a glassfish container was installed, so that the certA web application and some testing Attribute Provider could be deployed. This virtual machine is confined to a restricted intra-network within the AMA and can only be access by AMA employees therefore, since the purposes of this implementation was for testing, not all of the desirable security measures were implemented, in other words: HTTP was used instead of HTTPS, the certA’s certificate was not emitted by the EC CC, the used TSA was a testing TSA from the university “Instituto Superior Técnico” not the National TSA and the Signing Applet was not signed by a trustable certificate which made it impossible for it to be executed in a normal browser, instead it was launched like a standard java application.

In this intra-network there was also a testing version of the current Portuguese CC’s IdP, customized to use certA instead of its current Attribute Provider. The CC’s IdP uses a custom XML based protocol to communicate with its current Attribute Provider and since changing the IdP’s code was harder than changing the certA’s, some changes were made in order to certA understand the IdP’s attribute request. New java classes were added and a new java servlet was created, so that certA could now receive and understand the IdP’s requests.

With these changes and two testing Attribute Providers working with XML files playing the role of real databases for roles/attribute verification purposes, a testing small scale implementation of what could be working on the internet on pair with the National IdP was created.
For validating purposes a testing Service Provider built to test the National IdP was used to verify the Role Authentication Scenario and the Signing Applet was used to test the Digital Signature Certification Scenario. The validation process ended with the consent, of certain AMA representatives, that the solution delivers what is expected.

**Conclusions:**

Through the success of the case study and validation this thesis tries to reinforce the investigation hypothesis of the positive importance of Roles in Business Processes already defended by Professor Artur Caetano in its work [6] that fundamentally says that the necessity to model the real world onto the electronic one is continuously growing and doing so without the use of roles would lead to redundant, “full” of irrelevant relationships and difficult to organize world.

Several solutions where studied to achieve a successful association of qualified information to a person/user’s identity, though the use of digital signature formats like the AdES family turned out to be best suitable solution since it can answer to this thesis’ requirements by “only” requiring a trusted third party that can verify the authenticity of requested qualified data and sign the data has being verified and certified.

Some open standards where analyzed and conclusions to what standard to use where drawn in the related work. The final result lead to a solution involving the use of standards such as: SAML, XAdES and DSS with an EPM Profile.

The eID field is still growing and some form of standardization or best practices guide is need in order to have interoperability between foreign countries, the STORK project is an attempt to do just that but there are no guides in how to implement such a solution at the National level. This role certification topic is something that every country/entity interested in eID must take into consideration.

Once something like an “electronic identity” is created, it immediately awakens in the user the necessity to have an “electronic role(s)”.

Currently this work has some limitations and needs some future work to be developed:

One of this solution’s main limitations is the fact that it does not offer a back office for role and attributes management. In reality most of the Attribute Providers probably have their own management system, but the fact is that there might be some role/attributes that could be under the certA’s responsibility and therefore having no back office to manage them is a limitation.

This back office is part of the future work and could either be a simple Content Management System that manages a database with the available roles/attributes and their relationships with the citizens, or it could be a more complex and generalized system.
This second approach would be a system connecting and managing all the roles/attributes of the government trusted AP’s, the certA’s and small entities that don’t have the means to have their own AP. In this system even a citizen owner of a company could log on to and add the entry that another citizen is employed in his company; a system administrator or an AP representative could log on to the back office and add the entry that a certain citizen has graduated and is now an Architect. This new approach is the desirable one, but for a first launching stage the first approach is easier and quicker to implement, progressively evolving to a better system like the second approach.

Note that this generalized back office would be one of the new services that would benefit from this solution’s services: role based web authentication.

The second limitation of the current implementation is related to the first. The list of roles/attributes that the Signing Applet shows is a hardcoded one, that means that every time a new role/attribute is approved by the government the Signing Applet would have to be re-compiled, also, in a real case scenario that list would most certainly be enormous and it would be confused for the user to go through all those roles/attributes just to choose only one or two. In a future work, the Signing Applet, would have means to request to certA the list of roles/attributes associated to the citizen, after its successful authentication, and show only the relevant roles/attributes to the user. Currently the certA can make this request to the APs but it does not accept that type of SAML request in order to prevent a misuse of its aggregation capabilities from the IdP’s part.

The third limitation of this implementation is the incompleteness of the DSS Server. For non-repudiation capabilities the DSS server should be either able to update the XAdES Document's signature or implement the DSS EPM profile, currently it is not capable of either.

The XAdES document certified by certA actually has a TimeStamp so that in the future the document can be proven to have been signed at the time it was actually signed, but the problem is the DSS Server doesn't support updating signature requests which is required in order to offer a long term non-repudiation capability and even if it did that would not be the best approach since it would require for the user to remember to update its document signature every time his citizen or certA's certificate is about to expire.

Has an alternative solution the DSS Server can implement the DSS EPM profile that states the use of receipts that store non-repudiation data for each document that the DSS server signs, this way the user only needs to certified his document and in case the document needs to be put to test, even if in the future the certificates used to sign the document are expired the document can be proved to be valid at the time it was signed, by requesting to the DSS Server for the verification of an existing receipt with the same hash has the document's hash, dated with the document’s signing date and with the same signer’s ID has the document’s. In short, part of the future work would be to implement the DSS EPM profile in the DSS Server component of certA.

Also part of the future work is the submission of the two new SAML profiles for the OASIS approval and their release to public.
Bibliography