Digital Signatures with Familiar Appearance for e-Government Documents: *Authentic PDF*

Thomas Neubauer, Edgar Weippl, Stefan Biffl
Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
Email: {neubauer, weippl, biffl}@ifs.tuwien.ac.at

Abstract— **Most e-government applications have to find a solution for simple, reliable, secure and authentic signing of official documents. Citizens need a simple way to verify the authenticity and integrity of an official document. Currently XML documents allow representing such documents. However, the XML format does not guarantee a definite visual presentation of the document (presentation problem). In this paper we describe a solution approach - so-called *Authentic PDF* - using PDF technology that fulfills the following key requirements: 1) A visual presentation that resembles the traditional style of an official document; 2) A visual representation of the signature value that does not change the document authenticity; 3) The option for the holder of an official document to restore the electronic version of the authentic official document from the visual representation of the document (e.g. printout); 4) The filtering of dynamic content. We implemented and evaluated different approaches in a feasibility study using a typical e-government document set. Results of the study are: PDF is suitable to meet specific legal requirements on a signature solution in combination with a smartcard; the method has proven to be reliable and support a sufficient level of security.**

I. Introduction

During the last years, e-government in Europe and specifically Austria has increasingly gained momentum and importance. In Austria, the so-called e-government offensive provided the basis for carrying out official duties over the Internet. The idea is to conduct administrative procedures easily, quickly, and without special knowledge of the field or concerning jurisdiction. In this regard, the Austrian citizen card can represent the interface between citizen, corporations, and public authorities. In order to facilitate the access to e-government applications and to reduce user inhibition, signature solutions should only be based on standards and programs which can be taken for granted on the side of the citizen and which do not result in extra costs (e.g.: Browser, Adobe Reader). Another precondition for the acceptance of e-government by the citizens is the guarantee of certain fundamental security objectives: authenticity, integrity and non-repudiation. Currently, in Austria XML documents are often used to represent official documents. The Austrian government provides a security architecture based on XML [21] that allows the signing of documents based on XML in accordance with Austrian Law. On the one hand this solution supports the authorities in implementing a clearly defined security standard using defined JAVA-modules [22]. On the other hand a software solution for applying qualified signatures in combination with a smart card (e.g., Austrian citizen card) is distributed for no charge to Austrian citizens. These components represent established solutions, which have proven to be effective for the signing of XML/HTML/Text documents during the last years. However, the XML-format does not guarantee a definite visual presentation of the document and raises the so-called presentation problem [20], [19]. To guarantee its value as evidence, the presentation of the signed document has to be unambiguous. Using XML the definiteness of presentation can not always be guaranteed, due to numerous existing standards in connection with XML [20]. Additionally XML requires the user to invest into a certain degree of basic knowledge in order to be used efficiently. In the field of e-government XML might be a considerable barrier for inexperienced citizens. Therefore, an alternative to the XML approach would be the use of PDF. PDF is a system independent data format for easy document transfer over the Internet and is used by individuals, companies and public authorities. Today, most companies and organizations have huge amounts of information published and stored as PDF documents. Signing documents such as invoices would be useful. Nevertheless, many companies do not sign their PDF invoices they send to their customers. To fight tax fraud governments will require electronic invoices to be signed. With unsigned documents, a customer can forge the amount shown on the invoice and commit tax fraud. The increased use of PDF documents, the wide-spread circulation of related tools and the manifold ways of misuse require the enhancement of existing signature solutions. In this paper we focus on the design, implementation and validation of different solutions for signing PDF documents with a digital qualified signature. Therefore we describe the so-called *Authentic PDF* that uses PDF technology and fulfills the following key requirements:

- A visual presentation that resembles the traditional style of official documents.
- A visual representation of the signature on the signed document. The challenge of this issue is the application of the signature value by using PDF syntax and without invalidating the authenticity of the document. This requirement is the precondition for the restorability of a document.
- The option for the holder of an official document to restore the electronic version of the authentic official document from the visual representation of the document

Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)
0-7695-2567-9/06 $20.00 © 2006 IEEE

(e.g. from a print-out of the document). The process of restoring a document must guarantee that based on the print-out of the original electronic document the ”original” hash-identical electronic document is re-generated.
- The filtering of dynamic content of PDF documents for creating a definite visual presentation.

We evaluate whether the implementation of a signature solution based on the actual PDF-Specification 1.6 is feasible. Further, we examine if a PDF signature approach is an adequate alternative to already existing XML-based signing solutions. Based on this evaluation the implementation of tools for digital signing of authentic PDF documents can be performed.

II. Requirements on a signature solution

The use of digital signatures allows the validation of data authenticity and integrity. However, in a lawsuit the intentions of both parties have to be proven as well; therefore it is essential to know what the user saw when she signed the document. With documents that contain dynamic content and data formats such as XML, the presentation of a document is not always clearly determined [24], [5]. If, for example, a signed template does not exist in the first place or an unsigned template is changed afterwards, different presentations of the same document can be created. Likewise, the use of different applications and the manner of interacting with the software can lead to an indefinite range of different interpretations on the side of the user. Spalka et. al. [27] propose to solve the problem of dynamic content by restricting the actions of active content or the use of a ”secure viewer”. Kunz et. al. [20] evaluate the question in which way the use of XML might contribute to a solution for the presentation problem. Their conclusion is that many restrictions have to be fulfilled to guarantee a determined presentation of XML documents.

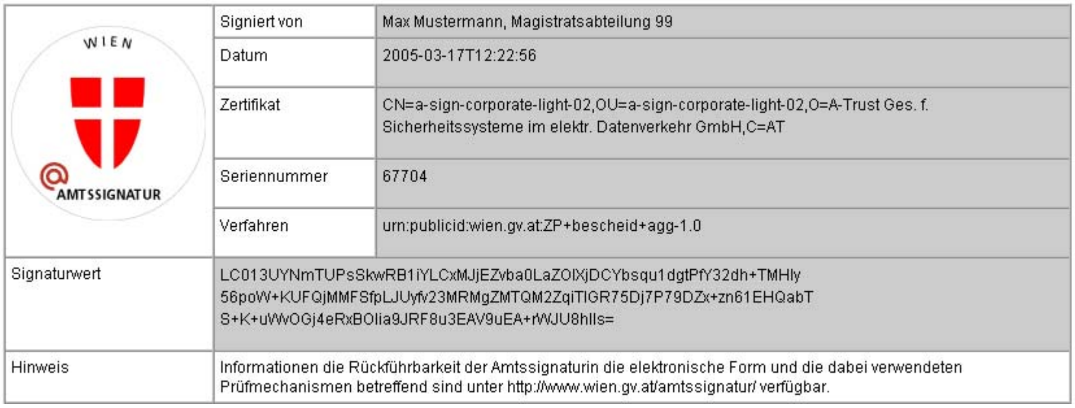
A definite presentation is especially important when ”non-repudiation” can not be completely guaranteed in the course of a legal trial e.g. due to an indefinite presentation. The precondition for ”non-repudiation” is that the signer was provided with mechanisms that guaranteed ”what you see is what you sign” [26]. Based on this basic paradigm Pordesch [24] explains the most important requirements on a signature solution:

- Definiteness of presentation of the data (to-be) signed.
- Transparency: The signatory must be provided with methods that guarantee that he sees what he signs. The verifier must be provided with methods that guarantee that allow him to see what was signed.
- Security: The presentation of the data to be signed must be correct, misinterpretations must be eliminated and manipulation of the used components must be impossible.
- Value as evidence: The descriptions of the presentation of the signed document must be part of the document and must be signed as well as the document.

Beside general requirements described above a valid signature solution additionally has to fulfill the legal standards in Austria. The Austrian Signature Law [9] and Signature Order [10] were adopted in January, 2000 and are directly based on the European Directive [12] on electronic signatures. However, the requirements of the Austrian Law are unique in the European Union [14], [13], [15]. The Austrian Law is very strict and requires higher security for the used technologies than the signature laws of many other countries. Thus, a system fulfilling the requirements of the Austrian Law is very likely to fulfill the requirements of other laws as well. Legal requirements include the visualization of the signature value on the document (so-called ”Authority Signature”) and the restorability of documents. These requirements are defined by the Austrian E-Government Act [8] as follows: The ”Authority Signature” is a signature applied to a document by a public authority, thus indicating that it is an official document. The depiction of certain features (e.g. official electronic seal, signature value) as part of the ”Authority Signature” facilitates a better recognition of the origin of the document, the verifiability of the signature, and therefore guarantees the validity of the document. The seal identifies a specific authority. The other fields contain the name and function of the signer, date and time, information about the issuer of the certificate, a unique ID of the document class, the signature value and a note about the restorability of the document into its original electronic form (Figure 1). The ”Authority Signature” may only be used by authorities. The verification of the signature must be possible by restoring the presentation of the whole document (e.g. print-out) into a form which allows the validation of the digital signature (electronic document). The evidential value of print-outs, which have been created based on documents with an ”Authority Signature”, is defined in conjunction with this property. Electronic documents from authorities printed out on paper, automatically claim the presumption of authenticity if the document has been signed with an ”Authority Signature” and the signature can be verified by restoring the electronic document. Additionally the Austrian law demands that prior to signing the document, the signer must have the opportunity to view a secure version of the document. To ensure that no changes are made to the document after the signature (e.g. by dynamic content such as Java) dynamic content must be filtered. Due to the fact that PDF documents can contain dynamic content, filtering the dynamic content is a precondition for achieving a definite presentation.

III. The PDF-Specification 1.6

PDF is a digital file format that allows the electronic display of a document’s visual presentation. The presentation is independent of the systems (software, hardware and operating system) that are used for creation, as well as of the systems needed for viewing the document. A PDF document can be created directly using PDF syntax, converted from other electronic documents, or digitalized from a paper document. The construction of a PDF document is defined by the following components [3]:



Signiert von	Max Mustermann, Magistratsabteilung 99
Datum	2005-03-17T12:22:56
Zertifikat	CN=a-sign-corporate-light-02,OU=a-sign-corporate-light-02,O=A-Trust Ges. f. Sicherheitssysteme im elektr. Datenverkehr GmbH,C=AT
Seriennummer	67704
Verfahren	urn:publicid:wien.gv.at:ZP+bescheid+agg-1.0
Signaturwert	LC013UYNmTUPsSkwRB1iYLCxMJjEZvba0LaZOlXjDCYbsqu1dgtPfY32dh+TMHly 56poW+KUFQjMMFSfpLJUyfv23MRMgZMTQM2ZqiTIGR75Dj7P79DZx+zn61EHQabT S+K+uWvOGj4eRxBOlia9JRF8u3EAV9uEA+rWJU8hlls=
Hinweis	Informationen die Rückführbarkeit der Amtssignaturin die elektronische Form und die dabei verwendeten Prüfmechanismen betreffend sind unter http://www.wien.gv.at/amtssignatur/ verfügbar.

Fig. 1. Example of an Authority Signature [7]

- Objects: A PDF document is a data structure composed from any given number of objects. There is a small set of basic types of data objects on the one hand and complex data types on the other hand.
- File structure: The PDF file structure determines how objects are stored in a PDF file, how they are accessed, and how they are updated. This structure is independent of the semantics of the objects.
- Document structure: The PDF document structure specifies how the basic object types are used to represent components of a PDF document. By that means, the semantic connections of the objects are defined.
- Content streams: A PDF content stream contains a sequence of instructions describing the presentation of a page or a graphical entity. The same object types and syntax as in the rest of the PDF document are used, restricted to basic types and direct objects, though. According to the PDF-Specification, content streams are interpreted and processed sequentially.

A digital signature is used to determine the authenticity of a document and its signer. The signature can be a method based on a PKI [2]. Starting with version 1.5 of the PDF-Specification, X.509 certificates [16] and the concept of PKI are supported. The realization of specific signature approaches as well as the definition of the needed signature methods is done by implementing a signature handler [3].

On a technical level, digital signatures in PDF consist of two components: A signature field (SigField) and a signature annotation (SigAnnot). These two components are created by the signature handler, once the user applies the signature. The signature field is a form field with the field value "signature dictionary". Like any other form field, the signature form field is also associated with another dictionary, the signature annotation dictionary, which includes the visualization parameters of the signature (appearance dictionary). By this means the appearance of the background, the name of the signer, the creation date and time of the signature can be defined. The specification "Digital Signature Appearances" [2] serves as a guideline for implementing the appearance of signatures in PDF documents. The administration of cryptographic properties of the signature is done by the signature dictionary (SigDict), wherein the signature handler (attribute: Filter) and the syntax of the encryption dictionary contents (attribute: SubFilter) are defined. The security handler is a software module (plug-in for Adobe Acrobat/Reader) that implements various aspects of the encryption process and controls access to the contents of the encrypted document. Third party developers are encouraged to implement security handlers of their own. Security handlers may use public-key encryption technology to encrypt a document. Examples of existing security handlers that support public-key encryption are Entrust.PPKEF, Adobe.PPKLite, and Adobe.PubSec. The SubFilter entry allows interoperability between handlers. A document can be decrypted by a handler other than the preferred one (specified by the entry 'Filter') if they both support the format specified by 'SubFilter' [3]. Public-key security handlers use the industry standard Public Key Cryptographic Standard Number 7 (PKCS#7) binary encoding syntax [25]. Apart from that, the range of data over which to compute the hash value (attribute: ByteRange) or other attributes, like reason or location of the signing, can be defined.

IV. Design and Evaluation of possible approaches

In this section, different approaches for realizing a signature solution using PDF will be discussed. In the course of this, the necessary extensions of and constraints on PDF will be defined. There will be a distinction between possible solutions for visualizing the signature value in a PDF document (Authority Signature) and for restoring the presentation of a document into its original electronic form. The design and evaluation of the different approaches is carried out according to the criteria on *Authentic PDF* defined in the introduction, the detailed legal requirements defined in section 2 and the PDF-Reference 1.6.

A. Authority Signature

This section presents possible solutions for the use case "Authority Signature". It focuses on the question whether the

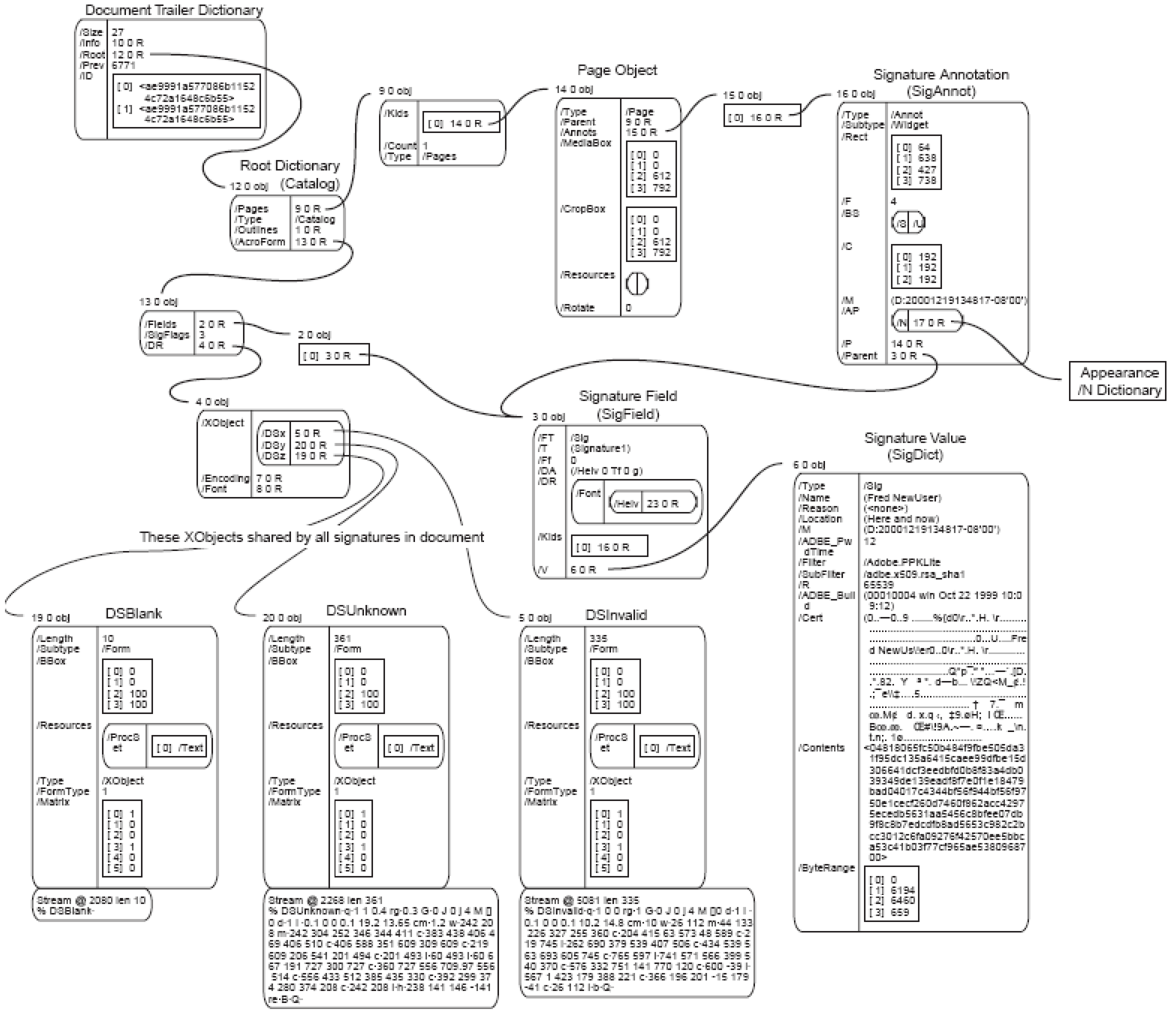


Fig. 2. Signature Objects in a PDF File [2]

implementation of the legal requirements on the ”Authority Signature” based on the PDF-Specification is feasible. For that reason, approaches which are based on the use of dynamic content, like Java or JavaScript, are excluded from this evaluation.

1) Creating two documents: In this approach, the official document is created using PDF. This document is signed, using a method which allows the verification of the signature with Adobe Reader. Then, a second document, in PDF or XML, is created. The extra document contains the content of the original document plus the signature value. It is not signed and only serves as basis for restoring the original document. The verification of the authenticity with Adobe Acrobat can only be done for the document without the visualization of the signature value. For restoring the original document, the other document, including the visible signature value has to be used.

2) Signing twice: In this case, extending the first approach, the document with the visualization of the signature value would be signed a second time. Therefore, the complete procedure can be described as follows: The designated PDF document is signed as usual. After that, the resulting signature value is visualized in the document, thus rendering the signature invalid. (For the sake of completeness, it has to be mentioned that by saving the data using an incremental update [3] of the PDF document, a rollback to the original signature and thus a verification of the authenticity would still be possible. Nevertheless, a user viewing the document with the Adobe Reader would be notified that the document has been changed.) Finally, the document is signed a second time, to allow the verification of its authenticity with Adobe Reader, even without having to do a rollback. In this case, the signature value would be displayed in the same PDF document. However, the data serving as the basis for the signature value (hash input data) and the data for computing the final signature would not be identical. Therefore this solution does not fulfill the legal requirements.

3) Embedding the signature value in the PDF document: This section describes options which are based on embedding the signature value in the PDF document. In contrast to the previously described options, the visualization of the signature value, as well as the signature itself, is placed in the same document. As the signature value is not created until the finalization of the signing process, it can only be visualized after signing the document. Therefore, the solution needs to provide a way to visualize the signature value on the document after finishing the signing process, without invalidating the authenticity of the document. Independent of the different implementation techniques explained in more detail afterwards, every specific approach meets the defined requirements of *Authentic PDF*.

a) Definition of two exclusions for the signature value: The visualization of the signature value can be realized by excluding two areas within the PDF document from the calculation of the hash value. On the one hand, this would be the attribute ”Contents” and on the other hand a second area with the same content as defined in the ”Contents” attribute, representing the visualization of the signature value. Both sections are excluded from the signature by setting the attribute ”ByteRange” appropriately. This method conforms to the PDF-Specification, which allows the exclusion of certain areas via the ”ByteRange” attribute. However, in the course of a concrete realization, there could be some problems because Adobe Acrobat/Reader allow only two entries in the attribute ”Byte Range” (= One exclusion). An individual signature handler is needed for validating the signature and both exclusions.

b) Visualization of the signature value via the ”Signature Appearance”: As already stated, it is possible to adjust the signature's appearance in a PDF document individually. For example, the author's own handwritten signature can be displayed in the signature field. Extending this concept, it is possible to show other data, which is present in the PDF document, in the signature field. By this means, it is also possible to display the signature value, which is stored in the attribute ”Contents” of the Signature Dictionary. The procedure starts with the user, who wants to sign the document, clicking on the pre-defined signature field in the PDF document. Thereafter, the password/PIN is entered and the signature value is generated. This value is then written to the ‘Contents' area of the PDF document and finally the signature appearance is created with the corresponding signature value. There are two possible variants:

- The process of creating the signature is defined according to the requirements needed for visualizing the signature value (for example by means of a special application). In this case, the signature value will be displayed in the signature block, just like the location or name attributes.
- ”XObjects” [3] can be used for visualizing the signature value. Thus, the visualization of the signature value is implemented the same way as the icons indicating whether the signature is valid or not (e.g.: green check mark).

Both of these solutions require at least an additional component such as a specialized application or plug-in in order to realize the described functionality. The ideal manner of implementing these approaches would be an extension of the standard features of Adobe Acrobat/Reader. That way, the signature's authenticity can be verified without needing any additional components.

c) Referencing the signature value: In this case, the visualization of the signature value in the PDF document is achieved by using an ‘active' field. This field would contain a reference to an area where the signature value is stored after signing the PDF document (usually the ”Contents” entry in the signature dictionary). Consequently, at the position in the PDF document where the signature value is meant to be displayed, a reference to the ”Contents” value has to be set upon creation of the PDF document. This method ensures that after signing the document, no further manipulations changing the hash value of the document and thus invalidating the signature are necessary,. Depending on the specific procedure for creating the PDF document, this entry could e.g. be generated by a special application. This approach represents a fairly simple option for visualizing the signature value. Unfortunately, the PDF-Specification currently provides no possibility to reference a single entry in a dictionary like that.

However, this approach can be extended in a way that allows the visualization of the signature value by means of a reference and therefore the verification with Adobe Acrobat/Reader. This can be achieved by defining the signature value as an object of its own, instead of storing it directly in the ”Contents” entry. Additionally two references to this separate object have to be set:

- One reference is set in the ”Contents” attribute: This provision guarantees that the document can be validated with Adobe Acrobat/Reader by using standard methods.
- The other reference is set at the position, where the visualization of the signature value is defined: By this means, the signature value will be displayed on the PDF document.

PDF documents of that kind can be created by using special applications. Depending on the applied signing methods the validation is generally possible with nothing more than the standard modules available in Adobe Acrobat/Reader. At the moment, one limitation of this solution arises from the fact that the signature value consists of hexadecimal characters, which would usually be displayed as special characters. Anyway, this problem could be solved by e.g. embedding a specific font.

B. Restorability

The concept of restorability refers to the possibility to restore or repeatedly create identical electronic documents where an ”Authority Signature” has been put on. With PDF documents, the process of restoration is much more complex than e.g. with XML documents. This complexity is mainly a result of the structure of PDF files and the existence of binary data in these documents. From a technical perspective the creation of PDF documents with an identical binary consistence is feasible. However, most approaches are connected with various constraints that limit their practical applicability. In the adjacent sections, several possible options are discussed. In this connection the following preconditions are assumed:

- PDF documents contain meta data like timestamps of the creation and modification date or unique identification numbers. This data has to be adjusted appropriately when restoring the original electronic representation of a document. For example, the timestamps in the PDF documents should be matched with the data visible on the presentation (print-out) of the PDF document.
- In the course of this evaluation, only the restorability of text documents will be examined, as the restoration of images, diagrams or tables is not possible, or at least not with reasonable effort.
- Usually, documents contain various styles of formatting (for example, headlines or other important passages of text are highlighted by bigger or bold fonts). The restoration of documents containing different formatting is generally possible as well, but much more complex. When entering the text for restoration, the text parts with specific formatting have to be marked in order to enable their electronic processing. This can be done, by assigning certain indications (like a tag in XML) or by entering them into a separate form field. However, as this problem also arises when restoring XML documents it will not be considered within this paper.
- The PDF-Specification does not contain standard methods for canonicalizing or transforming the reference data of a document into hash input data, for example, to sort out space characters. The necessary methods would have to be integrated into the applications which are used for creating the PDF files. The used methods must guarantee that the structure of the hash input data complies with the desired visualization in the PDF document. Anyway, the transformation or canonicalization can be carried out similar to XML, apart from certain PDF specific properties. For that reason, we refer to existing specifications [6], [28], [29].

1) Restoration by using PDF forms: In this scenario, the PDF document is created by using a PDF form. This form contains the static data of the document (e.g.: layout, certain text elements or pictures) and form fields. Upon creation of a document, the form fields are filled with the variable data (e.g.: name, address). The variable data can be retrieved directly from a database. When restoring the electronic document, the data stated on the print-out is e.g. entered via a web interface. After that, the data is transferred to the form fields of the PDF form. The process of restoration can then be finished by signing the document. In this approach the structure of the document is relatively restricted. Therefore the success rate of the restoration can be assessed as high. In the course of tests conducted with a typical set of e-government documents, it was possible to generate identical documents. Certain constraints are posed by the following set of data items, which are created and modified dynamically. All those entries are generated anew when saving a PDF file:

- Modification date
- ID: This entry represents a unique means of identification for the file (file identifier). This identifier is optional, but the PDF-Specification recommends using this entry, as omitting it might lead to processing problems in workflows which require a unique identification of the file.
- InstanceID: In principle, this entry serves the same purpose as the ID described above, but in contrast, it complies with the XMP-Specification [4].

In order to generate a file with an identical binary consistence, one solution would be to adjust these data items to the data of the original document. This can be achieved if this data is visibly printed on the document: the data is entered again when restoring the document (e.g.: the stated data (creation date) or parts of the signature value). Another possibility is to use dummy values for the entries, or leaving them out at all. The option of restoring a PDF document using forms has its reservations. Their scale depends on the different kinds of forms available. Theoretically, with the help of an extensive collection of forms, a wide range of use cases could be covered. Still, the following basic limitations exist:

- A mixed formatting of text within a single form field is not possible. By default, there can only be one uniform layout per form field, at least at the moment (e.g.: the complete content of a form field can be formatted bold or italic).
- Tables and images can only be used if they are already present in the base form.

2) Restoration by using special applications: This approach of restoration assumes that the PDF code is directly generated by a special application. Some of the limitations of the form-based approach could be eliminated by using this approach. The application has to guarantee that the objects of the PDF document are generated in a deterministic manner, so that a PDF document of identical binary consistence can be created. In order to examine the behavior, a number of tests were conducted with different text-based documents. For creating the PDF documents, the open source PDF generator iText [18] was used. In the first step, based on a text read in from a text file a PDF document was generated. In the second step, this document was signed. This procedure was carried out twice for every document, independently of each other. By this means, the restoration procedure was simulated. In the process of restoration the PDF document is also generated twice - once at the original creation and another time when restoring it. After every step, the independently generated documents were compared and evaluated in detail.

The most important result of the tests is the fact that it was possible with all test cases to generate hash-identical documents. The constraints are basically the same as in the form-based restoration approach. Consequently, creation date, modification date and identification numbers of the PDF document have to be considered in this approach, as well. Apart from that, it is important that the documents are created under the same circumstances. This requirement primarily comprises the process (algorithm) of creating a PDF document, all the components used (e.g.: fonts) and the methods needed for signing the document. Difficulties arise in connection with the use of diagrams and tables. Though, it would be possible to archive them, in order to be able to restore the identical PDF document at a later time. An alternative is the exclusion of those elements when computing the signature value. However, this option bears potential security risks.

3) Restoration by signing the document partially: A reduction of complexity could be achieved by partially signing the PDF document. In this approach only certain elements of the PDF document (e.g. the visible text) are signed. All the elements which are not restorable due to their complex structure (e.g.: diagrams or fonts) are simply excluded from the signature. The exclusion of certain areas when computing the signature value is technically feasible. However, the partial signing of a document bears potential security risks, since the authenticity and integrity of the elements which are not covered by the signature can not be guaranteed. Naturally, adding an object to the PDF document changes the hash value of this document. The situation is more or less the same when adding an indirect object by embedding a reference. Anyway, with partial signing of the document, the possibility that the signature remains valid although the document has been changed cannot be ruled out. In abusive intention, the byte range of the document could be defined only over the first byte of the document. In this case, all changes to additional elements of the document would not be covered by the signature, and thus remain hidden. In principle, this situation could be avoided by visualizing all the signed parts of the document. The advantage of this option would be that even the restoration of complex structures could be implemented, at least in theory. Nevertheless, the security risks of this approach have to be considered vis-a-vis its potential benefits.

V. Solutions for Filtering Dynamic Content in PDF documents

To ensure that no changes are made to the document after the signature (e.g. by dynamic content such as Java or JavaScript) and for providing a definite presentation dynamic content must be filtered. This section presents different options for filtering dynamic contents in PDF documents:

A. TIFF

In this scenario From a PDF file a TIFF graphic is created. It can be signed and transmitted. The drawback of this solution is that additional components for converting a PDF file to TIFF are needed. Moreover, no further automatic processing of the data is possible. There is an option to transmit another file (e.g. XML) with the data in addition to the graphics file. However, these data would not be signed or would have to be signed in addition to the graphic file.

B. XML

The PDF file is transformed to XML. More specifically, form data may be extracted and signed. This solution enables automatic processing of the data at the receiver. However, the drawback of this scenario is that the presentation of the document is not signed. Just as with the first option (TIFF), the XML file can be sent in addition to a file that contains the document view.

C. PDF/A-1

PDF/A-1 (ISO Standard 19005-1:2005 [17]) describes a file format designed for long term storage of documents. Documents may contain a combination of text-, raster- and vector-data. A compliant PDF/A-1 document has to follow the PDF reference 1.4 and is allowed to use all valid features unless the usage is prohibited or restricted by ISO Standard 19005-1. Furthermore, the PDF/A-1 standard defines how software tools are allowed to use the features when creating a standard compliant file. The usage of features, which are documented in earlier versions is prohibited. By using PDF/A-1, dynamic elements of a PDF file can be transformed and removed. Thus, the usage of PDF/A-1 fulfills the requirement of eliminating dynamic contents of a PDF file. Using PDF/A-1 to delete all dynamic content in a PDF file is primary recommendable when standard methods for storing a PDF file as PDF/A-1 file are available in e.g. Adobe Reader.

D. Virtual Print

Another option to create a PDF document without dynamic contents is the creation of a virtual printout of a PDF document. In this approach the dynamic content of the document is filtered automatically. Thereby the view of the original PDF document is transformed into another PDF document. For instance, Adobe's Distiller or Open Source projects such as PdfCreator [23] can be used.

VI. Conclusion

The use of PDF and PDF forms is a rather sophisticated approach for realizing a signature solution for public authorities and citizens. Although a PDF solution is significantly more complex than a XML solution, a technical implementation of the requirements for a PDF signature solution is feasible. This paper presented approaches for dealing with the given requirements. A visual presentation in PDF can resemble the traditional style of official documents and can be implemented. Additionally we showed that the visual representation of the signature value can be achieved without invalidating document authenticity. We implemented the visualization of the "Authority Signature" with a reference using standard PDF syntax. The main advantage of this solution is that the Adobe Reader can be used for validation. Further solutions using specific signature handlers were also proposed. The decision of using a specific signature handler depends on the signature methods used, e.g. the use of Elliptic Curve Cryptography [11] or other methods than provided by Adobe SDK [1] demands the implementation of a specific signature handler. Based on this requirement we presented solutions for restoring the electronic version of the authentic official document from its visual representation. We showed that the restoration can be realized either by applying a form-based approach or by using a special application. Additionally, the potential security risks of partially signing documents were presented. Moreover we showed that using PDF/A-1 would be an appropriate solution for filtering dynamic content of PDF documents. By filtering the dynamic content of a document it can be guaranteed that the presentation of the signed document is unambiguous. Therefore the use of PDF in general and PDF/A-1 in specific provides a solution to the presentation problem that can arise in connection with XML signatures. Other important criteria for a reasonable practical realization are its ease of use as well as the effectiveness and efficiency of the related workflows. This study has shown that there are several approaches for implementing a solution that is based on the PDF-Reference 1.6 and fulfills the very strict requirements of the Austrian law. In further work we will deal with the practical realization of a solution, which fulfills the identified requirements of *Authentic PDF*. We will focus on the analysis of potential security issues and conduct a quantitative evaluation of the proposed PDF-based signature solution approaches in terms of performance, complexity, and acceptability. Moreover, we will develop a concept for applying the shown approaches to scenarios apart from e-government.

References

[1] Adobe Systems, "Adobe Acrobat SDK." [Online]. Available: http://partners.adobe.com/public/developer/acrobat/sdk/index.html

[2] Adobe Systems, *Digital Signature Appearances*, May 2003.

[3] Adobe Systems, *PDF-Reference 1.6*, 2004.

[4] Adobe Systems, *XMP-Specification - Extensible Metadata Platform*, 2004.

[5] A. Alsaid and C. J. Mitchell, "Digitally signed documents Ambiguities and Solutions", *Proceedings of the 4th International Network Conference (INC2004)*, July 2004.

[6] A-SIT, "Entwurf Spezifikation Normalisierung von Freitext," Tech. Rep., 2004.

[7] "Amtssignatur (Authority Signature)" [Online]. Available: http://www.wien.gv.at/amtssignatur/

[8] "The Austrian E-Government Act: Federal Act on Provisions Facilitating Electronic Communications with Public Bodies, BGBl. I Nr. 10/2004."

[9] "Austrian Signature Law: Bundesgesetz ueber elektronische Signaturen (Signaturgesetz - SigG), BGBl. I Nr. 190/1999, BGBl. I Nr. 137/2000, BGBl. I Nr. 32/2001. (in German)."

[10] "Austrian Signature Order: Verordnung des Bundeskanzlers ueber elektronische Signaturen (Signaturverordnung - SigV), BGBl. II Nr. 30/2000. (in German)."

[11] I. Blake, G. Seroussi, and N. Smart, "Elliptic Curves in Cryptography", Cambridge University Press, 1999.

[12] "Directive 1999/93/EG of the European Parliament and of the Council on a Common Framework for Electronic Signatures." [Online]. Available: http://www.ict.etsi.org/eessi/e-sign-directive.pdf

[13] "e-Europe - Electronic Identity." [Online]. Available: www.electronic-identity.org

[14] M. Gentili, "Italian Electronic Identity Card - Principle and Architecture," *Proceedings of the 27th VLDB Conference, Roma, Italy*, 2001.

[15] A. Hayat, H. Leitold, C. Rechberger, and T. Roessler, "Survey on EU's eletronic-id solutions", 2004.

[16] R. Housley, W. Polk, W. Ford and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) profile," 2004.

[17] International Standard (ISO 19005-1), "Document Management Electronic document file format for long-term preservation Part 1: Use of PDF 1.4 (PDF/A-1)", 2005.

[18] "iText." [Online]. Available: http://sourceforge.net/projects/itext/

[19] W. Kubbilun, S. Gajek, M. Psarros, and J. Schwenk, "Trustworthy Verification and Visualisation of Multiple XML-Signatures", *Proceedings of the 19th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS2005)*, Springer-Verlag, Heidelberg, 2005.

[20] T. Kunz, U. Pordesch, and A. U. Schmidt, "XML-Signatur Anwendungsprofile als Weg zur Loesung des Praesentationsproblems," *Datenschutz und Datensicherheit*, 2003.

[21] H. Leitold, A. Hollosi, and R. Posch, "Security Architecture of the Austrian Citizen Card Concept," *Proceedings of 18th Annual Computer Security Applications Conference (ACSAC'2002)*, IEEE Computer Society, pp. 391-400, 2002.

[22] "MOA-Modules." [Online]. Available: http://www.cio.gv.at/onlineservices/basicmodules

[23] "PDFCreator." [Online]. Available: http://sourceforge.net/projects/pdfcreator/

[24] U. Pordesch, *Die elektronische Form und das Praesentationsproblem*, NOMOS, 2003.

[25] "RFC 2315, PKCS 7: Cryptographic Message Syntax, Version 1.5."

[26] K. Scheibelhofer, "What You See Is What You Sign - Trustworthy Display of XML Documents for Signing and Verification", *Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security Issues of the New Century*, Kluwer, 2001.

[27] A. Spalka, A.B. Cremers and H. Langweg, "Protecting the Creation of Digital Signatures with Trusted Computing Platform Technology against Attacks by Trojan Horse Programs", *Proceedings of the IFIP SEC 2001*, Kluwer Academic, Boston, MA, pp. 403420, 2001.

[28] W3C, "Canonical XML, Version 1.0," W3C, 2001.

[29] W3C, "Exclusive XML Canonicalization, Version 1.0," W3C.