Pervasive digital signatures: Syntactic robustness and simplicity of signed documents

Dimitrios Lekkas, Argyris Arnellos, Thomas Spyrou and John Darzentas

***Abstract*— The action of digitally signing has several intrinsic weaknesses that introduce syntactic and semantic distance between a signer and a relying party. As a result, digitally signed documents cannot be trusted and thus be widely deployed in pervasive environments. We evaluate the syntactic robustness of digitally signed documents by exploiting one key quantitative measure (the structural informativeness) and by comparing several qualitative characteristics of various alternative syntaxes. We are then able to identify which is the more reliable and simpler to transform syntax that will enhance the pervasiveness of signed documents, while it can be used in resource-constraint mobile devices. At the same time, digitally signed documents must preserve their security characteristics and their formatting and layout capabilities in order to achieve an enhanced level of trust on the semantic part of communication and thus be ubiquitously integrated with human users.**

***Index Terms*— Security, Syntactic distance, Objectivity, Informativeness, Novelty, Redundancy, Trust**

I. Introduction

A digital signature preserves basic security characteristics of digital documents, such as integrity and authenticity, while it is the principal verification of the signer's meanings as these are expressed in the respective signed document. Signing is an action which is always projected in the context of communication between a signer and a verifier. As such, signing acquires all the problems related to the indeterminacy of human communication, which are also intensified by its legal implications.

The term "ubiquitous computing" introduces a vision where technology disappears into the background. However, security poses fundamental challenges to realize this vision. The pervasiveness of digital signature applications is still very low and this fact may be credited to various intrinsic weaknesses of digital signatures that significantly reduce the acceptance, the usability and the trust against this technology.

The main procedure of creating and verifying digital signatures is based on the public key cryptography, where the signer encrypts (signs) a sequence of data using her private key and the verifier of the signature ensures the originality of the data by decrypting the signature using the public key of the signer and obtaining the original data [1], [2]. From the first steps of public key cryptography till nowadays, many value-added characteristics are enhanced, by integrating new technologies in the digital signature process. The hash algorithms gave a solution to the computational efficiency of the signatures, the digital certificates [3] and the self-certified keys [4] provided the means for effective identification of the signer, the Public Key Infrastructure (PKI) architectures build the necessary trust relationships and finally the time-stamping [5] and notarization techniques providing additional proofs that add value and longevity [6] to a digital signature.

The creation of a digital signature cannot be denied as an action, since it can be algorithmically proved, using cryptographic techniques. However, there are many weak points in the procedure of digitally signing data, since it is not performed directly by humans but only through hardware and software applied on binary data. Several questions arise such as who is using the signature-creation-data, whether she performs a willful act and whether the software and hardware used for this action can be trusted. Another important question is whether the signed binary data are uniquely transformed, displayed and observed by both the signer and the verifier of the signature (called the 'Relying Party' hereinafter) despite the fact that the integrity of the communicated data is guaranteed on the bit level. As a result, one may be held liable for a legally binding digital signature she created, without in fact having performed a conscious and willful act, due to ambiguities in the transformation and the presentation of the signed data.

Our objective is to identify the problem of syntactic (at the computational transformations and presentation level) and semantic (at the human cognitive level) distance between signed data, signer's meaning and relying party's understanding of this meaning. This distance consist a serious drawback for the pervasive usage of digital signatures, since it introduces fear and reduces the trust of the public against this technology. The paper is focusing on the syntactical characteristics of the signed documents that will affect their structural robustness, which in turn will designate the level of reliability of the communication between the two parties. We evaluate alternative syntactic techniques for communicating a signed document by measuring various quantitative and qualitative characteristics of these alternatives. We then propose the syntactic technique (format) that better mitigates

---

Manuscript received May 27, 2005.

D. Lekkas, A. Arnellos, T. Spyrou and J. Darzentas are with the Department of Product and Systems Design Engineering, University of the Aegean, Syros 84100, Greece (phone: 22810 97100, e-mails: dlek@aegean.gr, arar@aegean.gr, tsp@aegean.gr, idarz@aegean.gr)

the syntactic and consequently the semantic gap between the origin and destination of digitally signed documents, while it preserves its formatting capabilities in the simplest way. In other words, we introduce a basic framework towards ‘fair’ digital signing that provides even more favorable conditions for mutual understanding between the signer and the relying party and consequently a more pervasive usage of digital signatures.

II. The Action of Signing and Its Intrinsic Weaknesses

Signing is a personal action that is performed within a specific context or application. It attests a willful act by the signer and it is communicated and verified by one or more relying parties. Digital signatures can be used for security purposes and specifically to authenticate a document (i.e., to identify the signer), and/or to ensure the integrity of the document (i.e., to ensure that the binary data that consist the document have not been altered since it was signed). The action of the signer who creates a digital signature may signify the following:

1) Verification of Meaning – A signature evidences the signer’s meaning with respect to the document signed. The nature of the signer’s meaning will vary with the transaction, and in all cases can be approached only by looking at the context in which the signature was made. A signature may signify, for example, liability against an obligation, legal binding to the terms of a contract, the approval of a third party’s request, authorization to funds transfer, confirmation that the signer has read and reviewed the contents of a memo, an indication that the signer was the author of a document, or merely that the contents of a document have been shown to the signer and that she had an opportunity to review them.

2) Satisfaction of Legal Requirements – A signature is often used to satisfy a law or regulation that requires the presence of a signature before the document will be considered legally effective. European law, the electronic signatures directive [7], and other national laws, grant to the digital signatures legal validity equivalent to traditional hand-written signatures.

A fundamental intrinsic problem of digital signatures [8] is that the action of their creation (i.e. the display of a digital document and the usage of a private key) is not directly bind to a physical entity, but only indirectly through a machine and an application. The risk lies in the fact that the calculation of a digital signature is performed transparently by hardware and software (the signature-creation-device) that is mostly unknown and non-trusted for the end-user and that may be also malicious or at least unreliable. Risks may be identified in both the proper usage of the key and the objective notification to the signer of what exactly she is signing, known also as the issue of ‘What You See is What You Sign’ [10]. As a result, one may be held liable for a signature created by his private key on arbitrary data, without her full awareness or consent on this action.

In practice, there is a fundamental conflict between modern systems (including operating systems, applications and user interfaces) and security (in terms of protecting a secret key and securely present to the user what is being signed) due to the increased systems complexity and their reduced transparency. In other words there is no means to prove that the creator of a digital signature guarantees his awareness and that he performs a conscious and willful act. This fact is the basic weakness of digital signatures comparing to the hand-written signatures – which although are easy to forge, sometimes not-recognizable and not securely bind to one person – their creation is under the direct control of the signer and directly bind to a material (a piece of paper) that has a much more straightforward representation than a binary object.

The abovementioned weaknesses of digital signatures are directly related to the *secure pervasive computing*, in terms of *usability and trust (confidence)* in this security technology [9]. Summarizing, the issues of usability and trust are affected by the fact that digital signatures are not directly controlled by the signer, since:

- Signature is created by various APIs, interfaces and subsystems, not necessarily trusted.
- It is almost infeasible for a signer to create or verify a Digital signature by hand.
- Signature is calculated on binary data that may be differently interpreted and represented when creating a signature or when verifying a previously generated signature.

III. The Distance Between Signer’s Meaning and Relying Party’s Understanding

The action of signing is a purposeful action and as such it can be widely considered as agent-oriented. The signer is cognitive agent and therefore she has an intentional attitude providing her with a meaning towards a certain state of affair. She takes this meaning as information regarding that state of affair and wishes to communicate it to the relying party, so that the latter be aware for the signer’s meaning and therefore, about her intentionality towards this state of affair. The signer creates (or just reads) a syntactic component (e.g. a document in which she tries to express this information. Then, the signer is carefully interacting (reading) with the syntactic component to see if it properly (to a degree indicated differently by each different signer) expresses her meanings, that is, the way she relates her cognitive state with the respective state of affair. In case the signer is satisfied with this expression, and she wants to verify that this syntactic component can be used to provide her own information about that state of affairs, she signs the document.

One important issue that should be noted here is that the signer wishes to sign her meanings expressed in the syntactic component and not the syntactic component itself. However this is far from reality, where the signer just signs a series of bits. The whole procedure is in fact based on the *trust* of the signer that she shares the same (agreed) collection of symbols (alphabet) and rules of their arrangement (syntax) with the relying party and that the syntactic component is able to inform the relying party in a respective manner.

The objective is to reduce as much as possible the semantic distance between the signer and the relying party. The problem is concentrated in the global communication reliability of the syntactic component, which has to be evaluated on both the syntactic and on the semantic levels, while they are both affecting the meaning grasped by the signer and the relying party.

On the computational level of the application of a signature on a syntactic component, we identify that several transformation processes are introduced between the signer and the relying party, which as it has been shown, may destroy the integrity of the syntactic component, thus increasing each party's subjectivity. As illustrated on Figure 1, on the signer's side the binary representation of the component to be signed is transformed into a syntactic component with a well-formed structure (a known format such as HTML, PostScript or PDF) by means of various computational rules (alphabet, syntax, encoding, etc.) and presented to the signer through an analog device. The signer then decides to sign the syntactic component. The binary representation of the result of the signing procedure (a signed syntactic component) is again a series of bits, computed on the binary component, neglecting the series of possible transformations performed until their observation by the signer.

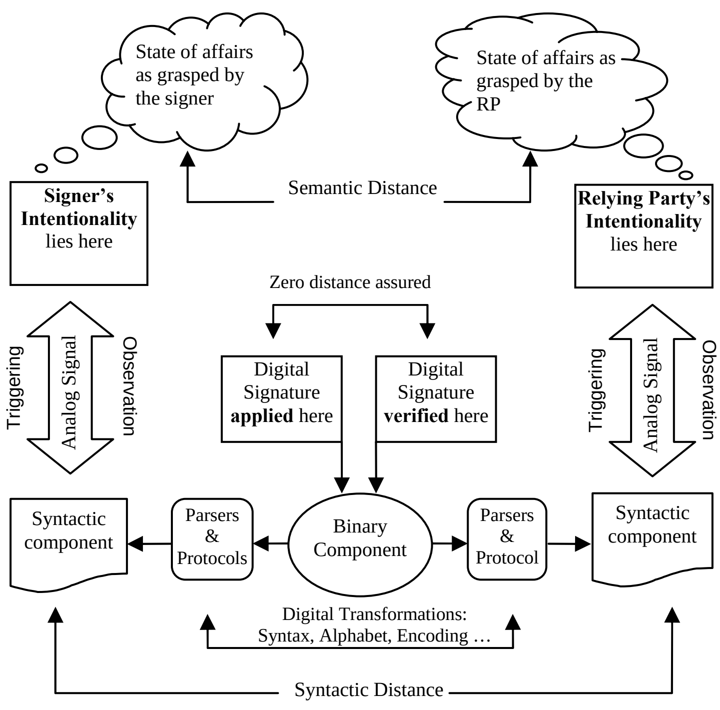

Fig. 1: The cumulatively added syntactic and semantic distance between signer and relying party

On the side of the relying party the signature is algorithmically verified against the signed binary component, together with some additional inspections such as the validity of the certificate of the signer, and the timestamp of the signature. Again, the verification is performed only on binary data that may be differently transformed, on the computational level, to the resulting syntactic component which will interact with the relying party. In other words, the digital signature only assures the integrity of the binary component (i.e. the two communicating parties share the same binary data) while it does not provide any assurance that the two parties identically transform the binary data on the syntactic and the semantic levels. We may now clearly see the cumulatively added distances on both edges of the communication, on both the syntactic and the semantic levels.

As a result of the long 'distance' between the signed syntactic component, the signer and the relying party, we may identify the following twofold problem:

1. A binary component that is signed on low-level (i.e. on bit level) may be differently transformed and thus observed as a different syntactical component, making the signer liable for a resulting meaning, for which he was not aware when creating the signature (*false positive*).
2. A simple alteration on the signed binary component, although it may not affect the observed syntactic component and does not alter the signer's intentionality, hence remaining possibly reliable regarding the relevant relying party's understanding, it renders the signature invalid (*false negative*).

Both views of the problem are crucial, since the first case erroneously imposes legal consequences, while the second case erroneously neglects previously imposed legal consequences.

A mitigation strategy that addresses both the problems of false positives and false negatives is to attempt to bind a digital signature to a syntactic component that will trigger a semantically identical result between the signer and the relying party, rather than to be bound up to raw binary data. Since, however, the creation of a digital signature is a clear algorithmic process solely performed on the binary component, a realistic objective would be the creation of a signature on data (binary component) that will be subjected to those transformational procedures providing a syntactic component with minimum ambiguity and subjectivity. Considering that the transformational procedures operate exclusively at the syntactic level, the problem is transferred (redirected) on the selection of those transformational procedures resulting in a syntactic component with the structural reliability that will better compensate the possible loss of the syntactic component's integrity. Consequently, such a syntactic component will have a structural capacity to inform the signer and the relying party with increased (greater) objectivity.

The issue of objectivity is directly related to two basic requirements of pervasive environments [9]:

- *The transparency of trust* that will enable the usage of digital signatures in everyday life and
- *The efficiency in resource-constrained environments* in the sense that a more straightforward syntax requires simpler and easier transformations.

IV. Exhibiting the Problem on the Computational Transformations Level

Before we try to analyse how the structural reliability of a syntactic component can be traced and evaluated, we give some examples that exhibit the basic problems of computational transformations on the syntactic level [11] [12] [13].

A. False positives: Documents with external references

Every digital component (even in its simplest form) has several external references such as encoding protocols, character mappings, formatting rules, dynamic content or image compression and transformation algorithms. This fact is the main reason that may lead to the correct verification of a digital signature even if the syntactic component representing the signed digital (binary) component may have many different and sometimes contradicting results. Some indicative examples follow:

Let's consider an HTML document with *external formatting reference* (cascading stylesheet) for the following scenario: Athena borrows €300 from Achilles, who in turn produces a maliciously written receipt in HTML and urges Athena to digitally sign it:

```
<html>
<head>
    <link rel="stylesheet" type="text/css"
    href="sign.css">
    <title>Signed Data</title>
</head>
<body>
   Athena owes<br>$
   <font class="color1">3000</font>
   <font class="color2">300</font>
   <br>to Achilles
</body>
</html>
```

The html is linked to an external cascading stylesheet, which, of course, is not included in the signed digital component. A maliciously written stylesheet includes the style 'font.color1' which makes the relevant text invisible:

```
Font.color1{visibility="hidden";
float="right"}
Font.color2{visibility="visible"}
```

Athena reads the syntactic component in Internet Explorer, as it appears in the first column of Table I and signs it (i.e. produces the signature value based on the underlying html code). Achilles then inverts the values of color1 and color2 in the stylesheet. The signature of Athena is still valid, but now Achilles claims that she has borrowed $3000 from him (2nd column of Table I). Even worse, Mozilla will completely ignore the value "hidden" in the stylesheet and will display the result (syntactic component) shown in the 3rd column of Table I, while the signature of Athena is still valid. The latter will also happen in IE, in case the external stylesheet is missing.

Table I: Different representations of the same HTML code

What Athena reads in IE when signing	*What Achilles presented in IE as signed document*	*What Mozilla displays. What IE displays when css is missing*
Athena owes $ 300 to Achilles	Athena owes $ 3000 to Achilles	Athena owes $ 3000300 to Achilles

The above example demonstrates also the problem of different manipulation of the same digital (binary) component by different applications (parsers) except of the linkage to external sources of encoding and formatting rules. Both problems in the resulting syntactic component seriously affect the trustworthiness of digital signatures.

Another indicative example of the false positive problem is the *character representation process*, which includes two major transformations: The character encoding and the glyph mapping. Figure 2 illustrates an example of possible character transformations that may lead the same binary component to be transformed to different syntactic components. The binary component that is equivalent to the decimal value 8805 represents the syntactic component 'greater or equal' according to the Unicode standard, but it is also equivalent to the decimal values 34 and 101 that represent the 'double quote' and the letter 'e' respectively. Furthermore, the correspondence of these values to a bitmap formatted syntactic component that can be displayed (the glyph) is also ambiguous, leading in some cases to not semantically equivalent results.

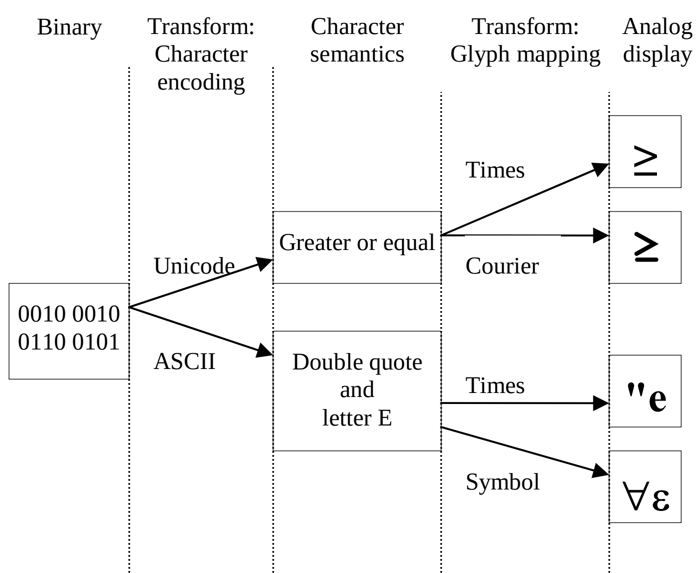

Fig. 2: Ambiguities in character encoding and displaying

Another characteristic of many digital document formats is the ability to include fields or portions of code that return *dynamic content* in the displayed syntactic component. System date, scripts that read additional binary components from other sources or system variables may render the displayed result unpredictable, although the original signed binary component remain the same.

B. False Negatives: Lack of canonicalization

Digital signature of an electronic document becomes invalid even if one single bit of this document is changed. However a change in the binary component does not always imply a change on the resulting representation of the syntactic component. In more extreme cases, some applications produce different BLOBS each time a document (syntactic component) is opened or printed (e.g. MS Office applications) and thus invalidate its signature without any change that is viewable or traceable by the reader. In any of the above cases, the signature should remain valid after a change in the binary component, thus avoiding false negatives. A method to ensure at some extend that the signed binary component will not change, is to use a 'simple' formatting syntax where its is possible to impose a 'canonicalization' procedure before signing. This procedure will eliminate any unnecessary or not well-formed layout or formatting information according to well defined rules.

As an example, the following two segments of HTML code, although typically different, will have exactly the same result on a display:

1	Hello World
2	Hello World

A digital signature applied on the first segment will not validate the second one, although it should be semantically valid for both cases. Such false negatives are very often as a result of simple document editing (but not altering) by various applications. A relying party, who does not need to be aware of the underlying protocols, would be rather confused in front of this case. A parsing of the above HTML code by a canonicalization procedure would ensure that only the second of the above equivalent codes would have been signed, thus avoiding false negatives.

As a counterexample for both false positives and false negatives we may view the case of *Code signing*. In this case, the identified problems on the syntactic transformations of the data do not stand, since the signer of the code of a program wishes to sign the binary component itself, rather than the syntactic component or the meaning it bears. This is due to the fact that a digital signature is used here for security purposes only, in terms of authenticity and integrity, while a relying party is not interested in the context of the code, its syntax or its semantic part. The executable code (binary component) also, is a machine-readable document that is not parsed by any transformation procedure.

V. Scoring the Structural Reliability of a Syntactic Component

We have seen that the signed syntactic component is the tool based on which the signer and the relying party will communicate regarding a certain state of affairs. Focusing on the fact that the transformational procedures affect the resulting structure of a syntactic component, what can be done is to find the structural characteristics of the syntactic component which are related to the possible quantity of information that can be created by the cognitive agent. This will give us the opportunity to enhance the structural reliability of the syntactic component and consequently the confidence level of the respective signed document in a pervasive environment.

A. Quantitative characteristics of a Syntactic Component: Structural Informativeness

In Shannon's information theory [14] (which would be better, as Floridi [15] among many others suggests, to be called as the Mathematical Theory of Communication), there is an intuitive connection between the information conveyed by an event (a message in a context of communication) and the surprise generated when such an event occurs. Specifically, the information conveyed in a message is inversely related to the probability of occurrence of this message. Usually, the probability of occurrence is interpreted as the unexpectedness of the receiver regarding that message, or her uncertainty before she receives it.

It should already be obvious that in Shannon's theory information can only be defined when there is both a sender and a receiver. However, it doesn't deal at all with the semantic aspect between them, that is, the meaning that a message may raise in the sender or the receiver. On the contrary, it is a purely quantitative approach to the technical problem of communication, namely in the quantitative definition of correctly transferring, as much as possible symbols, in an as fast as possible rate, from the sender to the receiver, via a given communication channel. To do that, Shannon needed a quantitative measure of the amount of information contained in a sequence of symbols. Therefore, assuming that there is a sender and a receiver and the former is a binary source producing a number of symbols, before the symbols be communicated, the latter would have an uncertainty as she wouldn't know which symbol the device would have produced. Thus, the amount of units of information produced by a sender S communicating a message M from a set of messages consisting of N equiprobable messages, equals the number H of binary decisions needed in order to select a particular message from them. Formally, we may say that

$$H = log_2 N$$

In that case, and being in accordance with the intuitive connection between the *'informativeness of a message'* (meaning the average information or the expectation value of the information content of a single symbol) and its unexpectedness regarding the receiver, the prior probability of occurrence of each of the N symbols is equal to $P=1/N$. Thus, based on the additive property of the quantitative measure of H, we can say that the information content I_i of the i^{th} message of a source S, with prior probability P_i and $\Sigma(P_i)= 1$ is given by the equation:

$I_i = -log_2 P_i$

It is now obvious that in this framework, the lower the prior probability of occurrence of a message, the higher is the information content of its occurrence. The quantity I_i provides the novelty value of the specific message [16].

We can now generalise that for a binary source producing messages consisting of N symbols with prior probabilities of occurrence $\{P_1,\ldots P_n\}$, where $\Sigma(P_i)= 1$, the average informativeness (*meaning the average information or the expectation value of the information content*) of a message M is given by

$$H = -\sum_{i=1}^{N} P_i \log_2 P_i \text{ (bits per symbol)}$$

As it is argued in [16], this can be said to be the measure of expectation value of the *novelty* content of the symbol of a source. It can be implied that in case the structural units (symbols) used to construct a message posses equal prior probabilities of occurrence, then, the average information content (informativeness) of the source constructing a message equals the measure of $H = log_2N$ bits/symbol (used in the message).

B. Computing the Informativeness of Known (Document-based) Transformation Protocols

Focusing on the fact that the applied transformation procedures transform the binary component into a formatted syntactic component, we proceed to compute the informativeness of various transformation (formatting) protocols. This will give us a measure of the informativeness of a respective syntactic component, which is a measure of its structural capacity to inform, or, in other words the richness of the formatting capacity of the document. Although a 'rich' formatting capability provides a better tool for communication between the signer and the relying party regarding a certain state of affairs, this is not always the best choice in terms of security. In fact, assuming that the analogue result of various syntactic components is the same, the lower the informativeness of the document is, the more reliable is the communication. In other words, it is more secure in terms of transformation integrity to pass the same output using the simplest protocol.

Specifically, working on the level of symbols we consider that a particular formatting protocol has an alphabet of formatting symbols (e.g. markup tags) plus an alphabet of verbal content symbols (e.g. the characters of Latin alphabet). For each type of document consisting of N formatting and content symbols we may compute the probabilities of occurrence for each symbol and the average structural information (informativeness) contained in this formatted document, based on the equation of the previous section.

As a case study, we have chosen to compute the informativeness of eight document-based formatting protocols, being: Four text-based protocols (plain-text, HTML, XML and RTF) two binary formats (PDF and MS-Word) and two image formats (Bitmap and JPEG). We have converted some documents (mainly with formatted verbal content, which is the usual case for digitally signed documents – e.g. the present paper) into all the above formats, assuring that their analogue representation looks (almost) the same, except, of course, of the plain-text document.

For the plain-text document the counted symbols are the 26 Latin characters plus some punctuation symbols. For the text-based documents with formatting capabilities, we counted the Latin characters of the content part, plus the formatting symbols, being the distinct <> tags for HTML and XML or the strings between two backslashes (or a backslash and a space) for RTF. For the case of a bitmap image, we assumed that in an 8-bit color depth image the symbol (formatting and content) is a pixel, whose color is represented by an octet of bits. Thus, we counted each distinct octet in the bit stream as a symbol (i.e. maximum 256 different octets). Counting the presence of octets (lacking of any better measure) in the bit streams of the other binary formats (i.e. PDF, MS-Word and JPEG) these symbols proved to be rather equiprobable (i.e. rather random) and therefore the value of informativeness was computed at a much higher value, as expected. The results of the case study are summarized in Table II:

TABLE II: THE INFORMATIVENESS OF DIFFERENT DOCUMENT FORMATS

Document syntax	**Distinct Symbols (N)**	**Total Symbols (S)**	**Informativeness (H)**
Plain-text	91	28552	3.0814
XML	149	29499	3.2124
HTML	173	29776	3.2473
RTF	468	35721	3.8578
PDF	254	153814	5.3118
MS-word	254	168312	5.8532
JPEG	254	72089	5.5069
Bitmap	174	381214	1.5674

A text document with formatting and layout capabilities represented as a bitmap image, has the lowest informativeness. From the text-based format (excluding plain-text which has no formatting capabilities) XML and HTML have the same low informativeness.

C. Qualitative characteristics of a Syntactic Component

Focusing on the transformation procedures applied on the binary component, we may identify several qualitative characteristics of various transformation protocols that affect the structural reliability of the relevant syntactic component and consequently their confidence level in a pervasive environment.

The parameters taken into consideration are divided in four basic categories:

1) Readability on the semantic level:

- *Formatting and layout capabilities*: Documents (with verbal content) capable to represent text formatting, structuring and layout can give a 'richer' analog representation than their plain-text equivalents, providing better communication semantics.

- *Existence of meta-data*: The intrinsic capability of the protocol to include customized meta-data within its signed part, is a positive characteristic. For example the inclusion of the type of the document, the protocol and the version used and other descriptive information increase the objectivity of the transformations. Additionally, the existence of meta-data adds a '*predefined logic'* between the communicating parties that will reduce the semantic distance of the exchanged messages.

2) Readability on the syntactic level:

- *Low Complexity*: In the context of the present analysis we define complexity in terms of human readability. We consider a protocol having low complexity when a human can perform the basic transformation process and follow the results without using a computational system (e.g. this stands for an HTML document), while a high complexity protocol refers to binary documents (e.g. image BLOBS or PDF) where it is practically impossible for a human to reproduce the result. According to this definition, a low complexity (human-transformable) document exhibits much more objectivity, since it can be more easily trusted by humans (signer and relying parties).

- *Existence of canonicalization rules*: Canonicalization acts complementary to a transformation protocol, imposes the construction of well-formed documents and contributes to the elimination of false negatives in digitally signed documents (see the example in section IV.B)

3) Low Novelty

- *No Dynamic content*: Documents that include dynamic content produced by non-deterministic code or scripts that display arbitrary results within the document (e.g. system time) also increase ambiguity.

- *Publicity and standardization*: A public, widely available and standardized protocol gains advantage (in terms of objectivity) against unknown proprietary protocols.

4) High Redundancy

- *No External references*: As illustrated in the examples of section IV the usage of (not signed or standardized) external references such as style-sheets or character encoding protocols increase document ambiguity and may result to unexpected results.

- *Embedded transformation protocols*: A document capable to include the parser or the transformation protocol within its body before it is signed, reduces ambiguity.

Aiming to evaluate the overall objectivity of the selected protocols, we score each protocol positively, negatively or neutrally against each of the abovementioned parameters, as illustrated in Table III. In detail, plain-text documents lack any formatting capability, while all other protocols are capable to give specific structure and layout to the data. Readable and custom meta-data can exist only in XML structures (e.g. XML-signature standard) and in HTML headers. In respect to external references, plain-text and images are the only formats which cannot include external references, while, for example, XML and HTML may refer to style-sheets and PDF or MS-Word formats may refer to fonts, to other files and to system variables. We may find dynamic content in HTML (assuming that the usual parser of an HTML document is a web browser supporting script execution) in RTF, in PDF and in MS-Word. We consider PDF and MS-Word as proprietary protocols while all others are public and standardized. The only protocol that permits the embedding of transformation rules (e.g. character encoding and font representation) is PDF. Canonicalization applies only to XML and HTML documents. Finally, in terms of complexity, we consider that the protocols that can be parsed by a human are plain-text, XML, HTML and partially (score 0) RTF and small bitmaps.

TABLE III: SCORING THE QUALITATIVE CHARACTERISTICS OF TRANSFORMATION PROTOCOLS

	Total Score	Formatting & Structuring	Meta-data	No External References	No Dynamic Content	Publicity & Standardization	Embedding	Canonicalization	Low Complexity
Plain-Text	1	-1	-1	+1	+1	+1	-1	0	+1
XML	4	+1	+1	-1	+1	+1	-1	+1	+1
HTML	2	+1	+1	-1	-1	+1	-1	+1	+1
RTF	-2	+1	-1	-1	-1	+1	-1	0	0
PDF	-3	+1	-1	-1	-1	-1	+1	0	-1
MS-Word	-5	+1	-1	-1	-1	-1	-1	0	-1
Bitmap	2	+1	-1	+1	+1	+1	-1	0	0
JPEG	1	+1	-1	+1	+1	+1	-1	0	-1

D. Towards robustness and simplicity in pervasive environments

Based on the qualitative evaluation of the transformation objectivity (Table III) and on the measurement of informativeness (Table II) of the various document formats we reach the result shown in Figure 3.

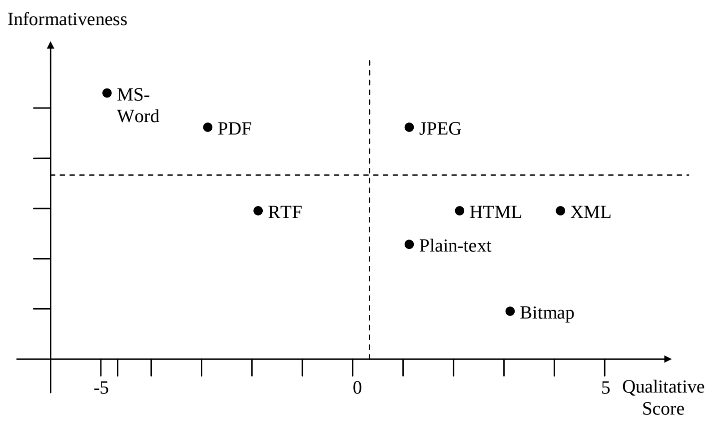

Fig. 3. Evaluating the overall syntactic reliability of documents

Towards our objective which is the enhanced syntactic robustness and simplicity of digitally signed documents, leading to higher pervasiveness, we may now assume that:

1. The higher is the score of qualitative characteristics measured, the higher is the *objectivity* of the transformation procedure and the most straightforward is the representation of signed data, thus resulting to *less false positives*.
2. The lower is the *informativeness* of a document type, the less are the redundant symbols and the simpler is the syntax, thus resulting to *less false negatives and false positives.*

Clearly, the syntaxes in the lower-right quarter of the chart (Figure 3) are most suitable for digitally signed documents in pervasive environments. Excluding plain-text documents, which have no formatting capabilities, bitmap images and markup languages (XML and HTML) proved to be the simplest, more reliable and easier to transform syntaxes for digitally signed documents. As a result, the *usability* and the *acceptance* of digitally signed documents will grow, leading to their enhanced pervasiveness [17].

VI. Conclusions

Digitally signing cannot be denied as an action, since it can be algorithmically proved, using cryptographic techniques. However, there are many weak points in the procedure of digitally signing data, since it is not performed directly by humans but only through hardware and software applied on binary data. One emerging question is whether the signed binary data are uniquely transformed, displayed and observed by both the Signer and the Relying Party, despite the assured integrity of the communicated bits. This situation may lead to false positives, rendering one liable for a legally binding digital signature she created, without in fact having performed a conscious and willful act. On the other hand, a legally binding signature may be neglected or even denied, due to a small alteration on the data, which does not necessarily affect the communicated semantics (false negative). The above weaknesses of digital signatures constitute a serious drawback for the usability and the acceptance (confidence and trust) of this technology that negatively affect their usage in pervasive environments.

The informativeness of a document is a measure of the probability of occurrence of the symbols within the document. This is interpreted as the novelty or the richness of a document in respect to its syntactic capability to inform. Assuming that we can use several syntactic alternatives to produce the same analogue result, the one with the lowest informativeness is preferred for signed documents, since it reduces complexity and enhances the readability on the relying party's side. Other qualitative measures that affect positively the syntactic robustness of a signed document are its human readability on the syntactic and the semantic level, the low novelty and the high redundancy. The above measures are also connected to the metric of informativeness, which proved to be a key value indicating the syntactic robustness of signed documents.

The evaluation of the above metrics, as a case study, showed that the document syntaxes based on mark-up languages (XML and HTML) or plain bitmap images are highly preferred for applying and verifying digital signatures. Since these formats exhibit high syntactic reliability, they can be widely trusted, accepted and used and consequently they must be considered as the only alternatives in pervasive computing.

References

[1] Rivest R.L., Shamir A., Adleman L., "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, Vol.21 No.2, pp.120-126, 1978

[2] Menezes A.J., van Oorchot P.C., Vanstone S.A., *Handbook of Applied Cryptography*, CRC Press, 1996

[3] Kohnfelder L., *Towards a practical public-key crypto-system*, Thesis, MIT, 1978

[4] Girault M., "Self-certified public keys", in *Advances in Cryptology: Eurocrypt'91, LNCS 547*, Springer-Verlag, pp. 490-497, 1991

[5] Adams C., Cain P., Pinkas D., Zuccherato R., "Internet X.509 Public Key Infrastructure Time-Stamp Protocol", *IETF Request For Comments 3161*, available at http://www.ietf.org/rfc/rfc3161.txt , 2001

[6] Lekkas D., Gritzalis D., "Cumulative notarization for long-term preservation of digital signatures", *Computers & Security*, Vol.23 No.5, pp.413-424, 2004

[7] European Union Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 1999

[8] Maurer U., "Intrinsic limitations of digital signatures and how to cope with them", in Proceedings of the *6th Information Security Conference (ISC'03), LNCS-2851*, pp.180-192, 2003

[9] Thomas R.K., Sandhu R., "Models, Protocols, and Architectures for Secure Pervasive Computing: Challenges and Research Directions", in Proceedings of the *2nd IEEE Conference on Pervasive Computing and Communications Workshops*, PERCOMW'04, pp.164, Orlando, Florida, 2004

[10] Sceibelhofer K., "What You See is What you Sign – Trustworthy display of XML documents for signing and verification, In *Communications and Multimedia Security, CMS'01*, pp.3-13, Darmstadt, 2001

[11] Alsaid A., Mitchell C.J., "Digitally Signed Documents – Ambiguities and Solutions", in Proceedings of the *4th International Network Conference (INC2004)*. July 2004

[12] Josang A., Povey D., Ho A., "What You See is Not Always What You Sign" in proceedings of *the Australian UNIX User Group, AUUG'02*, Melbourne, 2002

[13] Sveda P., "Trustworthiness of signed data", FIMU Report Series, Faculty of Informatics, Masaryk University, available at:

http://www.fi.muni.cz/veda/reports/files/2002/FIMU-RS-2002-06.pdf, Sep 2002

[14] Shannon C. E., Weaver, W., (1998), "*The Mathematical Theory of Communication*", Urbana and Chicago, IL: University of Illinois Press

[15] Floridi L., "Information", In *The Blackwell Guide to the Philosophy of Computing and Information*, edited by Luciano Floridi, 2004

[16] Küppers, Bernd-Olaf, *Information and the Origin of Life*. Cambridge, MA: MIT Press, 1990.

[17] Scholtz J., Consolvo S., "Toward a Framework for Evaluating Ubiquitous Computing Applications", *IEEE Pervasive Computing*, Vol. 3, No. 2, pp. 82-88, 2004