Meaningful electronic signatures based on an automatic indexing method

Maxime Wack, Ahmed Nait-Sidi-Moh*, Sid Lamrous, Nathanael Cottin

Systems and Transports Laboratory

University of Technology of Belfort-Montbeliard

90010 BELFORT Cedex

France

{maxime.wack, ahmed.nait-sidi-moh, sid.lamrous}@utbm.fr, nathanael.cottin@ensicaen.fr

ASTRACT. Legal information certification and secured storage combined with documents electronic signature are of great interest when digital documents security and conservation are in concern. Therefore, these new and evolving technologies offer powerful abilities, such as identification, authentication and certification. The latter contribute to increase the global security of legal digital archives conservation and access. However, currently used cryptographic and hash coding concepts cannot intrinsically enclose cognitive information about both the signer and the signed content. Indeed, an evolution of these technologies may be necessary to achieve full text researches within hundreds or thousands of electronically signed documents. This article aims at describing a possible model along with associated processes to create and make use of these new electronic signatures called “meaningful electronic signatures” as opposed to traditional electronic signatures based on bit per bit computation.

1. Introduction

The EESSI final report (EES, 1999), of which the European Parliament and Council 1999/93/CE directive (EPC, 1999) inspires, validated the technical and juridical acceptability of an electronic signature linked with a digital content. This electronic signature would be considered, depending on security and environmental criteria, as a strict equivalent to a manuscript signature. Since March 13, 2000 and the French law 2000-230 (OJ, 2000), a legal value can be attributed to electronically generated, signed and stored documents as long as they meet common criteria, such as fidelity and long-time conservation, listed by the French Civil Code.

As far as electronic signature applied on numeric information is concerned, professionals express their wish to take part of this necessary enterprise documents management evolution, especially in terms of signature processes and signed documents archival and retrieval. We thus propose in this article a possible model which handles electronic signatures construction in accordance with the legal context previously mentioned (Hansz, 2006).

Electronic signature has become a widely used way for authenticating digital information. It benefits from numerous researches on asymmetric cryptography and hashcoding.

Basically, electronic signature reproduces old wax seals used in the antiquity (NIST, 2000). A seal may be compared with a secret signature key which should be in the sole possession of the signatory (the entity that signs up the information). However, a main difference between seals and electronic signature must be pointed out: whereas a seal is affixed on the support of the information (the paper for example), electronic signature is generated using the information itself. As a consequence, a seal remains the same (which makes it possible to identify its proprietary) but an electronic signature depends on the information it refers to and therefore is different from one another. As a consequence, a given source information must always produce a unique

* Corresponding author, Email address: ahmed.nait-sidi-moh@utbm.fr

electronic signature and two different sources must lead to generate two different electronic signatures (using a single hashcoding algorithm along with a single private key, called signature key). To each signature key corresponds a unique public key, called verification key. This verification key verifies the signature authenticity and the information integrity (we suppose that the hashcoding process is collision-resistant). The identity of the signatory cannot be proved until a legally approved link can be given between the verification key and the signatory, such as a public-key certificate.

Electronic signatures generation relies on asymmetric cryptography[1] applied on hash coded data. Contrary to common cryptography, electronic signature does not aim at preserving information confidentiality but rather data authenticity and non-repudiation (Kaeo, 1999).

2. Motivation

2.1. A new hashcoding approach

Hashcoding (Menezes, 2001) which creates digital fingerprints from data flows generates a fixed-length message from any source flow. The result is independent from the source flow length. Let $h(\ )$ be a secured hashcoding function used on a source flow s. To be eligible for traditional electronic signature construction, this function must answer the following requirements:

- A given source flow s must always produce the same hash code.
- Two different source flows s and s' must produce different results (strong collision resistance)
- A hash code value does not hold any computable information which could be used to recreate its source flow (non-reversibility).

These requirements may be mathematically expressed as follows:

1. $h(s)=h(s') \Leftrightarrow s=s'$ and $h(s)\neq h(s') \Leftrightarrow s\neq s'$.
2. $h(s)=d \Rightarrow p\left(h^{-1}(d)=s\right)\rightarrow 0$, where d refers to the digital fingerprint produced by $h(s)$ and p is a probability. Indeed, the second logical equation indicates that although reconstructing s from d is theoretically possible, it appears to be computationally infeasible.

As these algorithms directly work on bit flows (and are destructive), they do not provide ways to get a rough view of the hashed source data.

In the particular case of pure textual documents (and mainly taking legal considerations into account), it may be of great importance to get the main ideas of a document: the impact of the proposed technology is that one can grasp the original document ideas from its hash-code, which is not the case in the traditional hash functions where the relation between a document and its hash is not reversible. Lawyers find all their interest in the application of this method in order to find information related to a document starting from its fingerprint. A meaningful signature may advantageously supplement other document retrieval methods as it would increase the "recall"[2] level. As Losee et al. (Losee, 2003) claims, a high recall level "is desirable in environments such as law or academia where the cost of missing a relevant document may be very high".

Therefore, lawyers may want to consider electronic signatures as proof elements by themselves (e.g. when the signed content has been destroyed, either accidentally or on purpose), which traditional hashing techniques do not provide.

Thus, rethinking hash-coding, we suggest making use of document words to process the hash-code prior to creating the electronic signature (with help of a private key). Although our hashcoding technique may be used in place of traditional hashcoding systems using real numbers, we intentionally use integers as we consider that collision resistance is not of great interest (documents within a legal scope).

---

[1] Data deciphering or decrypting leads in getting comprehensible information from a ciphered text. However, decryption is used when the entity does not own the appropriate key and thus tries to break the ciphered text.

[2] This retrieval performance measurement indicator refers to the probability a document is retrieved given that it is relevant to a given search query.

2.2. Related work

In the framework of information systems, and in order to ensure the security of electronic interchanges, a certain number of hashcoding algorithms have been proposed and implemented, such as MD2 (Kaliski, 1992), MD4 (Rivest, 1991) and RIPEMD (Dobbertin, 1996) (Preneel, 1997).

For example in (Preneel, 1995) the security of message authentication code (MAC) algorithms is considered. In this study, a new generic construction is proposed for transforming any secure hash function of the MD4 family into a secure MAC of equal or smaller bit length and comparable speed. Rivest proposed two very fast hash functions which are motivated by their use for RSA data security, namely MD4 and MD5 (Rivest, 1992a, 1992b). As the successor to MD5, the SHA (Secure Hash Algorithm) family (NIST, 1995) is a set of related cryptographic hash functions. The SHA algorithms were designed by the National Security Agency (NSA) and published as a US government federal information processing standard. The SHA algorithms are used for the protection of sensitive unclassified information and encouraged to be employed by private and commercial organizations.

Some of the most widely used algorithms to generate electronic signatures (mainly SHA-1 and MD5) have been proved to be no more collision-resistant (Wang, 2005) and new versions have been submitted to the community.

As said previously, secured hashcoding algorithms are designed to make sure (in theory) that the original message cannot be reconstructed from its fingerprint. As this type of hash codes does not give valuable information on the source flow, it must be used when the hash code has to be transmitted to a third party that needs to know the existence of an information without knowing the content of this information (during a time-stamping process for example), or when the electronic signature is generated on a multipart document which includes multimedia content (images, sound, video) and text formatting elements (extra tags use to format the document on screen or for printing).

In particular, such electronic signature is computationally but not semantically linked to the source message. We suggest hereafter a new hashcoding technique which applies on textual documents and based on words spatial representation (Lamrous et al, 1997).

3. Meaningful electronic signature creation process

Electronic signature is of great importance as far as the signed content refers to a juridical context. However, the WYSIWYS (« What You See Is What You Sign ») paradigm is limited when the “to be signed” document encloses a dynamic content (macros or dynamic fields for example), holds secrets (by means of images steganography techniques) or needs a non-secured (i.e. non-approved) proprietary viewer. Our scope is therefore strictly limited to textual information (which makes the most of juridical documents).

Associating a semantic with the electronic signature allows to make sure that the source document and the generated electronic signature are strongly linked and also that the source data used to create the signature are solely made of static textual elements (without any visualisation or printing information, etc.) which do not need any particular viewing process.

Generating this kind of electronic signatures allows to efficiently retrieving information from archives with limited full-text (time and processor consuming) researches.

3.1. Indexing with hash-code

Whereas common electronic signatures generation processes rely on bit-related hashcoding, we define the notion of trace. A trace gathers all lemmas[3] along with a set of calculated parameters that provide information on words distribution and position within the “to be signed” document (Lamrous, 1999). This trace dramatically reduces the size of the original document and one can roughly recreate a document from its trace. Traces can also be compared to determine the correlation between documents. It may be possible to get confidence in the

[3] Lemma : common part of a word which can be declined to produce related words : « worker » and « work » are both issued from « work » lemma

similarity of two documents in analyzing their first lemmas: the more identical parameters values, the more confidence in their similarity, considering that a trace is unique and collision-resistant.

As a trace may enclose many parameters (and thus may not be applicable to electronic signature generation for time-consumption and size of the generated trace before the cryptographic process), we have worked on reducing the size of our traces on the one hand and keeping significant meaning on the other hand. Therefore, we focused on lemmas found to be meaningful by Lamrous algorithm (Lamrous, 1999), followed by their traceability (reduced set of parameters which do not denaturize the distribution of these lemmas within the document). This retained process allows constructing a certain number of signatures varieties, independently from languages and domains.

Finally, to simplify the trace generation process, an extra step is performed before the trace algorithm applies: all accentuated characters (such as in French language) are replaced by their non-accentuated equivalent ($\acute{e} \Leftrightarrow \hat{e} \Leftrightarrow \grave{e} \Leftrightarrow e$) and special characters are removed (parenthesis, percentage, etc.). The whole content is then lowercased to avoid case-sensitive comparison tests.

A list of parameters, called traceability, is computed for each significant lemma. It contains the following fields:

- The lemma occurrence or pertinence degree of the lemma.
- Its barycentre.
- Its variance.

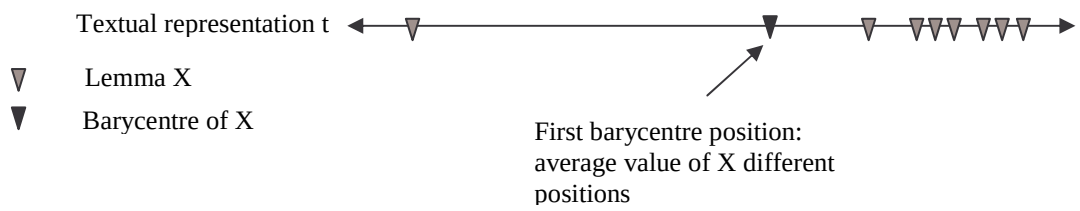
3.1.1. Lemma distribution, barycentre computation

We suggest focusing on the textual spatial distribution signal of each lemma. However, keeping an exhaustive list of every lemma would increase the trace size. Indeed, a representative value of this list, called barycentre, is created instead. A barycentre can be described as a harmonic centre within the whole textual representation of the document. This metric value varies from 0 to 100: a near 0 value means that the lemma appears most at the beginning of the document and a value which approaches 100 indicates that the lemma is concentrated at the end of the document. A lemma with a level-heading of 50 is quite ambiguous as it may be interpreted as a uniform distribution or as a concentration around the centre of the document. The variance of the lemma helps the interpretation: it is used to measure the coverage rate of the lemma within the document as far as it evaluates the distance of the lemma declinations (known as the user question): the smaller this distance, the more the connection between the subject and the question.

The positions numbers (ranges) of the lemma are considered as basics source data. To uniform results, digits are converted into a 0 to 100 value in comparison with the size of the document. Let “X” be a lemma with an occurrence value of 1. It is located at position 36 in a document which comprises 120 words. The computed position equals $(36 \times 120)/100 = 43.2$. This computed value refers to the lemma location within the textual representation of the document.

Hereafter is a graphical example which illustrates a barycentre computation for a given lemma having an occurrence greater than 1 (they appear more than once in the document). The purpose is to associate an average but representative value of the location of this lemma in the document.

Let t be a textual document represented by an axis graduated from 0 to 100 (barycentre minimum and maximum values). A lemma distribution is represented on this axis as follows:



The previous figure illustrates the particular case of a word positions average value which leads to represent an inappropriate vision of the barycentre. Indeed, the single left-side value can generate a passivity of the barycentre. It appears thus suitable to assign different multiplicative coefficients depending on the average value

of the standard deviations: a linear levelling which depends on the distance between this average value and the punctual positions has to be performed.

Therefore, let:

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\partial = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2}$$

Where:

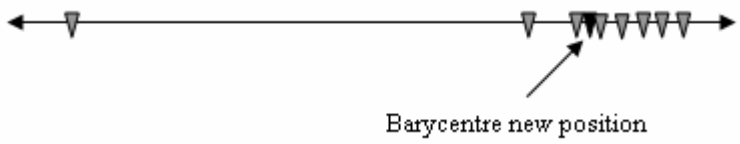
- x_i : x (word) different positions for every i value from 1 to n
- n : occurrence value of the X lemma

One uses a Gaussian function that associates low weights with distant points with $f \in [0, 1]$. More precisely, more x (word) is close to the first average, $f(x)$ must take a great value (close to 1) and conversely.

The corresponding barycentre value can then be calculated:

$$Barycentre = \frac{\sum_1^n f(x_i)x_i}{n}$$

Base on this barycentre value applied to lemmas, the new gravity centre is shown hereafter:



3.1.2. Pertinence degree of the lemma

We experienced the curve expressing lemmas occurrences in function of their corresponding barycentre values, as depicted by figure 1.

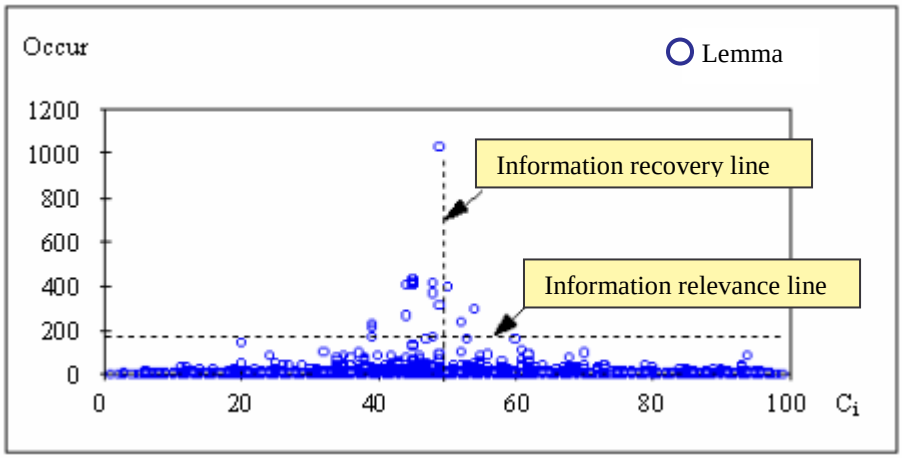


Figure 1. *Lemmas occurrences in function of their gravity centre within a given document textual representation.*

One can point out the symmetrical view of the curve considering a vertical median called “information recovery line”. Relevant lemmas are numerous and located near this bar: this is the case of the “smoke” lemma, which is uniformly quoted. Therefore the document is more likely to deal with smoke (and its derived words, such as smoking).

Other noticeable numerous lemmas located far from the vertical median represent local pieces of information. Finally, a particular point appears at the top the information recovery line. It represents the “of” lemma of the English language. The combination of its highly levelled value and its close location to the vertical median make it non-relevant (or “empty”): it is necessarily a frequently and uniformly employed word. An “empty words” class may thus be defined: it encloses empty words such as “the”, “of”, “it”, etc., usually grouped around the median and high-levelled (upper the information relevance horizontal line).

The baseline of the curve indicates lemmas mentioned once (frequency equals 1), called *hapax*[4], which is the most common case.

The vertical and horizontal lines (information recovery line and information relevance respectively) are used as an entry point for calculating the gradual level-heading of words within a document. The median is fixed whereas the relevance line vale depends on the number of lemmas extracted from the text. An empiric approach has been proposed by Lamrous (Lamrous, 1999) from an experimental corpus. The latter was constituted of very different textual representations of numerous documents, taken from a variety of domains and of all sizes. However, all of these documents were written in the French language. Generalities can be derived from this experimental work and expressed as an algorithm which can determine the relevance degree of each lemma in accordance with the whole text based on the spatial representation of figure 2. A similar model must be applied for any other language.

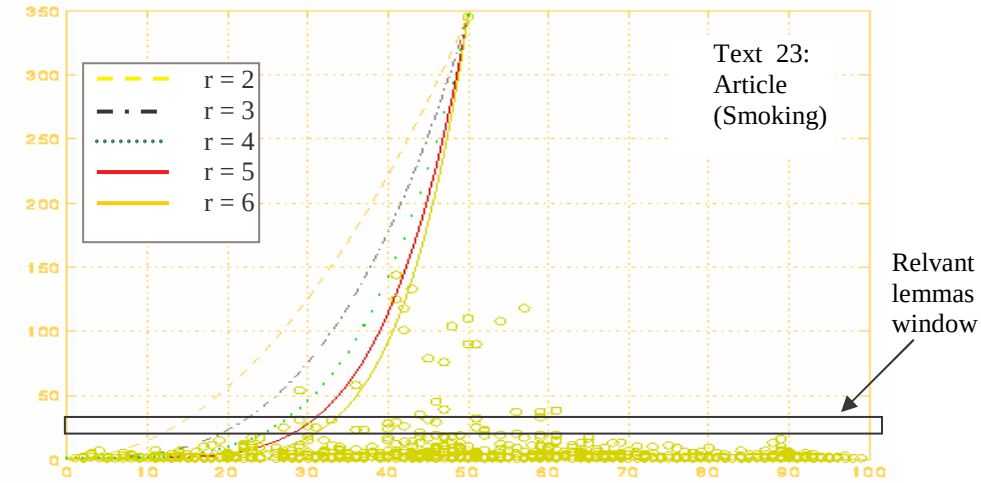


Figure 2. *Spatial representation example of an article dealing with smoking.*

The relevance degree varies from 0 to 100. Any 100 valued lemma is necessarily located within the relevance window determined using the algorithm proposed by Lamrous. These lemmas are used to generate the hash code value. This hash value is then computed to create the electronic signature value.

It is crucial to take care of the lemma pertinence degree compared to the corpus in order to refine the fourth parameter (relevance degree). Let’s take the example of a given lemma, say “concept”, being relevant but repeated and scattered in a set of documents. This lemma becomes globally irrelevant and is considered as an “empty” word because of its lack of discrimination power. Moreover, our working context makes it tricky to index all lemmas based on a corpus which gathers all documents corpuses due to the multiplicity of the languages used. This parameter may thus cause ambiguities. In our specific case of generating meaningful electronic signatures, the fourth parameter computed by the algorithm seems sufficient to provide cryptographic computation with a meaningful fingerprint created from a list of text-based (contrary to corpus-based) relevant lemmas.

3.1.3. Working process evaluation

This experiment relies on the analysis of a textual example. Our goal is twofold: it consists on the one hand in evaluating the automatic relevance degree attribution process, and pointing out the lexicometric characteristics of the text on the other hand.

[4] From the Greek expression “hapax legomenon” which means “thing said once”

Our text proposal is an excerpt of a medical journal. It focuses on smoking and related cardio-vascular diseases.

Here are some indicators:

Total number of words of the text: 3848

Number of different words: 908

Highest appearance degree: 226

The following table[5] shows the results obtained on a French document which deals with smoking when the algorithm is applied on significant words:

A: Lemma.

B: Lemma appearance degree.

C: Lemma barycentre.

D: Word variance.

A	B	C	D
cigarette	14	30	761
infarctus	11	49	361
maladie	23	42	441
egalement	15	44	484
atherogene	13	25	484
atherosclerose	13	25	484
myocarde	12	39	484
myocardique	12	39	484
important	17	48	529
effets	11	19	625
facteur	20	36	625
deces	11	66	625
hyperten	15	73	729
effet	20	55	729
montre	11	48	784
etudes	14	45	841
fumee	12	44	841
vasculaire	18	28	900

Table 1. *Uncompressed trace results: 405 octets.*

The techniques described before intervene in our meaningful electronic signatures generation process. Classical conservative ("without loss") compression algorithms, our trace (see table 1) can be encoded to produce a sequence or binary values. The global size of the trace may also be reduced (less than 405 octets). This sequence of '0' and '1' bit values, called "encoded trace", is used as a hash code value input in the signature generation process. Contrary to common hashcoding algorithms, our encoded trace has a variable length depending on the size of the document (and consequently the number of extracted lemmas). However, one can extrapolate that the representation of the encoded trace size versus the size of the source document reaches an asymptotic line. Therefore, our traces may not become very heavy and remain acceptable for our signature generation purpose. *The evaluation of the asymptotic line coefficient and benchmarks are currently in progress and will be soon published.*

[5] This table has not been translated as words frequency differs from one language to another

3.2. Integration into electronic signature

Back to general considerations, even if our hashcoding process reflects a rough overview of the signed content, it does not provide any trusted information about the signatory. A non-refutable link between the signature (private) key (respectively the corresponding verification – public – key) and an identifier of the signatory must be presented to the electronic signature verifier. One of the most widely employed standards, called Public Key Infrastructure or PKI (Adams, 1999) relies on a pyramidal infrastructure which testifies this link in providing signatories with a signature key along with an electronic certificate.

This certificate basically mentions the signatory identity (first name, last name), contact information (home address, home phone number, work company, work address, work phone number) and the signatory's verification key (which can be publicly distributed).

This information is sealed by the Certificate Authority (CA) that generates and delivers the signatory's certificate. This certificate is sealed with the issuing CA's electronic signature. Trust in the certificate information consists in trusting the signing CA and the certificate current state (see figure 3).

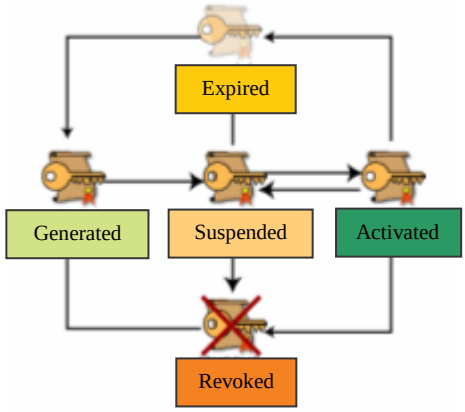


Figure 3. *Digital certificate life cycle.*

As each certificate remains active for a limited period of time (it mentions beginning and end of validity dates) and may be revoked (in case of loss or compromising), the verifier must make sure that:

1. The issuing CA is recognized as a trusted CA third party by courts of law.
2. The issuing CA's signature key is not compromised.
3. The signatory's certificate is valid and in its activation period (neither suspended nor revoked).

Once the verifier gets confidence in the certificate information, the signatory's verification key can be taken out and the electronic signature verified.

The use of our algorithm is reflected into the internal certificate structure. Figure 4 is an example of an X.509v3 certificate (Housley, 2002). The signature algorithm of the certificate proprietary (signatory), such as "SHA1withRSA" indicates that the Lamrous algorithm is employed instead of SHA-1 (in this particular example). The issuing CA signature algorithm used to seal the certificate is not modified as the owner's public key sequence of bits must be signed along with identity and validity (textual) information.

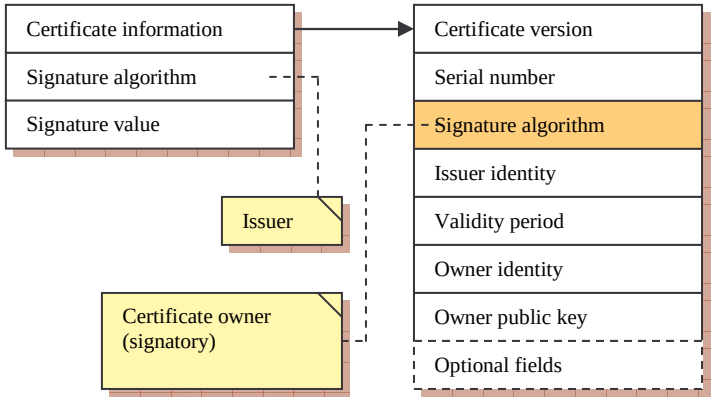


Figure 4. *X.509v3 certificate internal structure.*

4. Evaluation

4.1. Trace word vector relevance evaluation

We voluntary apply different types of noise onto a given word of the trace. Let's take the example of "cigarette" from table 1 to demonstrate that:

- Our method is suitable to produce collision-resistant hash-codes.
- Similar documents traces are very close: this property is useful when one need to prove plagiarism on documents protected by law even when the original document has been lost or destroyed.

	Occurrences	Barycentre	Variance
Original text	14	30,88	761,88
Noise #1	14	30,92	767,41
Noise #2	14	30,90	762,48
Noise #3	15	32,47	745,28
Noise #4	13	29,67	816,68
Noise #5	14	30,87	761,27

Table 2. *"cigarette" barycentre and variance vectors depending on various noise types.*

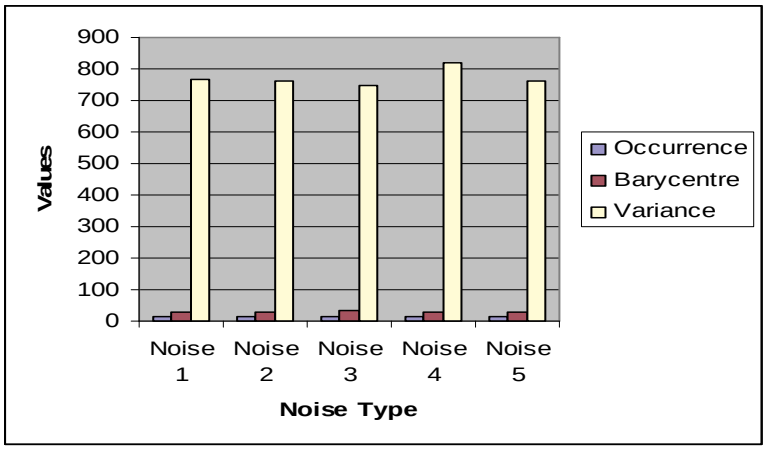


Figure 5. *A simulation result.*

4.2. Documents similarity measurement

The interpretation of documents traces by lawyers may raise the plagiarism question. In order to quantify the similarity degree we suggest the following calculation method:

Let D_1 and D_2 be two documents very close to each other in terms of trace value (and thus make use of the same relevant words).

We associate to D_1 and D_2 two triplets vectors $V_1 = \{o_1, b_1, v_1\}$ and $V_2 = \{o_2, b_2, v_2\}$ where $\{o_i, b_i, v_i\}$ respectively refer to the occurrence, barycentre and variance values of document i such that:

- Occurrence is the average value of all words occurrences which take part of the trace.
- Barycentre is the average value of all words barycentres which take part of the trace.
- Variance is the average value of all words variances which take part of the trace.

The similarity is obtained as follows:

$$sim(D_1, D_2) = \frac{D_1^t \times D_2}{\sqrt{(D_1^t \times D_1) \times (D_2^t \times D_2)}}$$

$$= \frac{\sum_{i=1}^{3} D_1^i \times D_2^i}{\sqrt{\sum_{i=1}^{3} \left(D_1^i\right)^2 \times \sum_{i=1}^{3} \left(D_2^i\right)^2}}$$

where D_j^t is the transpose of the vector D_j.

A "near 1" similarity value refers to a slight difference between these two documents and thus a small angle value. D_1 and D_2 are then very similar.

5. Conclusion

We suggested in this article a method for generating secured hash codes compared to "classical" hash coding techniques based on bits computation (such as MD5 or SHA). This method which relies on the relevance degree of selected words within a textual document allows creating a "meaningful" trace used as a fingerprint. This trace is a reduced representation of the text it refers to and can thus offer a rough view of the document content.

Such hashcoding method is suitable in terms of size and computation performances and thus appears eligible for electronically signing legal documents. It may substitute to existing hash coding methods on textual documents. One can appreciate the possibilities of this method as the generated trace becomes the signed document thesaurus. As archiving processes usually rely on detached signatures (signatures separated from the signed content), we are now able to pre-classify documents without knowing their exact contents but with help of the signature thesaurus. However, the proposed meaningful signatures should be used directly by data search engines as it would be very time consuming to "decipher" each signature to get its enclosed thesaurus. Indeed, it may be valuable to keep this thesaurus next to the signature to perform data search on the thesaurus. Once a document appears to be relevant (based on given search criteria), it is selected along with its signature.

This integration within archival trusted third parties processes as well as a deeper evaluation of our hash-coding process and archival system performances will be discussed in another communications.

7. Bibliography

Adams C., Farrell D., *RFC2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols*, March 1999.

Dobbertin H., Bosselaers A., Preneel B., *RIPEMD-160, a strengthened version of RIPEMD*, Fast Software Encryption, LNCS vol. 1039, D. Gollmann Ed., pp. 71-82, 1996.

European Electronic Signature Standardization Initiative (EESSI), *Final Report of the EESSI Expert Team*, July 1999.

European Parliament and Council 1999/93/CE directive.

Hansz B., A. Nait-Sidi-Moh, M. Wack, S. Lamrous, N. Cottin, “Signature Signifiante”. Submitted to the European Office of Patents. Patentlaan 22280 HV Rijswiijk (ZH), 2006.

Housley R., Polk W., Ford W., Solo D., *RFC 3280: Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile*, April 2002.

Kaeo (1999) : M., 1999. Designing Network Security. Macmillan Technical Publishing, USA, ISBN 1-57870-043-4.

Kaliski Jr B. S., *RFC 1319: The MD2 Message-Digest Algorithm*, RSA Laboratories, January 1992.

Lamrous S.-A., *Modélisation et réalisation d’un système prototype interactif de recherche d’information multimédia à forte composante textuelle*, phD thesis of the University of Technology of Compiegne, 1999.

Lamrous S.-A et Trigano P., *Organisation des bases documentaires, vers une exploitation optimale*, revue Document Numérique, Hermes ed., Volume 1 – n° 4/1997, pp. 441-458, 1997.

Losee R. L., Church Jr L., *Information Retrieval with Distributed Databases: Analytic Models of Performance*, IEEE Transactions on Parallel and Distributed Systems, pp. 18-27, 2003.

Menezes (2001) : Menezes A. J., Van Oorschot P. C., Vanstone S. A., Février 2001. *Handbook of Applied Cryptography.* CRC Press, USA, ISBN 0-8493-8523-7.

National Institute of Standards and Technology (NIST), *Secure Hash Standard (SHS)*, Federal Information Processing Standards Publication, FIPS PUB 180-1, April 1995.

National Institute of Standards and Technology (NIST), *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication, FIPS PUB 186-2, January 2000.

Official Journal, March 14, 2000, p.3968.

Preneel B., Paul C. and V. Oorschot, *MDx-MAC and Building Fast MACs from Hash Functions*, proc. Crypto’95, Springer-Verlag LNCS, August 1995.

Preneel B., Bosselaers A., Dobbertin H., *The cryptographic hash function RIPEMD-160*, CryptoBytes, vol. 3, No. 2, pp. 9-14, 1997.

Rivest R. L., *The MD4 message digest algorithm*, proc. Crypto’90, LNCS 537, Springer-Verlag, pp. 303-311. 1991.

Rivest R. L., *RFC 1320: The MD4 Message-Digest Algorithm*, MIT Laboratory for Computer Science and RSA Data Security, April 1992.

Rivest R. L., *RFC1321: The MD5 message digest algorithm*, Internet Activities Board, Internet Privacy Task Force, April 1992.

Wang X., Yu H., *How to Break MD5 and Other Hash Functions*, Advances in Cryptology, Eurocrypt'2005, Lecture Notes in Computer Science Vol. 3494, R. Cramer ed., Springer-Verlag, 2005