A Novel Watermark Algorithm for Integrity Protection of XML Documents

Ronghua Yao, Qijun Zhao, and Hongtao Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, P. R. China

Summary
With the fast development of Extensible Markup Language (XML) and its comprehensive application, the integrity protection of XML documents is becoming pressing. A traditional method for this is digital signature. In this paper, however, based on watermark techniques, we propose a novel solution to the integrity protection of XML documents. In this scheme, watermarks are generated through applying Principal Component Analysis (PCA) on a matrix constructed from an XML document and a key. Then they are embedded into the XML document by altering the case of letters in the XML tags according to the Exclusive OR (XOR) results of the XML document and watermarks. To testify the integrity of a watermarked XML document, watermarks are generated again and the original document can be exactly retrieved from the XOR results of these watermarks and the watermarked document only if it is not tampered; otherwise, the retrieved document will be an illegal XML document. Compared with existing signature methods, this scheme is excellent in its simplicity and economy of storage and channel bandwidth. Experimental results also demonstrate that our proposed watermarking scheme is less time-consuming than the traditional signature methods.

Key words:
Extensible Markup Language (XML), Integrity Protection, Watermark, Signature, Principal Component Analysis (PCA)

1. Introduction

Today XML (Extensible Markup Language) is widely used in information exchanging [1], Web Service [2], Business-to-Consumer (B2C) and Business-to-Business (B2B) applications [3], etc.. In fact, World Wide Web Consortium (W3C) has designated XML as a standard of representing Web data [4]. Meanwhile, security of XML documents has become one of the most pressing concerns about XML applications [2][5][6]. Take Web Service, believed to be promising in the Web-based applications, as an example, security is the bottleneck of its further development. Galbraith et al. gave survey on this issue [2] and the latest progress in this field can be found in [6]. Among various security problems, integrity protection, the focus of this paper, is one of the most significant ones.

Existing methods to protect the integrity of XML documents are based on the signature technique [2][5][7]. W3C has proposed a standard signature for XML documents [8]. However, such methods increase the workload of storage as well as bandwidth required for transmitting messages. In real applications, it is of great importance to reduce the cost in both time and space, especially when the information to be dealt with and transmitted grows explosively. Watermark techniques [9][10] are recently developed as effective tools for both copyright protection and integrity protection. Such techniques embed the protecting information, namely watermarks, in the original documents rather than attaching it to them as the signature technique does. Thus it is believed that watermark is superior to signature if efficient algorithms are proposed for its generating, embedding and verification. Although a number of watermarking algorithms have been proposed for digital images [11], video [12] and audio [13], only a few are reported for text documents because there is very little redundant space in them for watermark embedding.

Essentially, XML documents as well as HTML (Hyper-Text Markup Language) documents are plain text documents and thus it is much more difficult to embed watermarks in them. Katzenbeisser et al. [9] proposed to embed watermarks by adding space and tag into the source code of HTML web pages (Space-Tab Coding, or STC, hereafter). However, STC also has the problem of expanding file size. Zhao and Lu [14] have proposed another watermarking scheme for the tamper-proof of HTML Web pages. Through altering the case of letters in HTML tags, they embed watermarks into HTML documents successfully and avoid increasing the file size of documents. In this paper, we extend Zhao and Lu's work to protect the integrity of XML documents. Compared with existing signature-based methods, the proposed watermarking scheme is much more efficient in terms of consumed time and free of the embarrassment of increasing file size.

The remainder of this paper is organized as follows. In section 2, we give a brief review of XML and the digital signature of XML documents according to the W3C standard. After introducing related works, the proposed XML watermarking scheme is presented specifically in section 3. Then, section 4 shows the experimental results. This paper is finally concluded in section 5, where we also give future research directions.

Manuscript revised January 2006.

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<SignedInfo>
 <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
 <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
 <Reference URI="">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>9438E169E6688163D38174837BB6933D0F32497B=</DigestValue>
 </Reference>
</SignedInfo>
<SignatureValue>D87C25AD2AACC5A717C5B8C7BFFAD77143768DBF=</SignatureValue>
</Signature>
```

Figure 1: An example of detached XML signature

2. XML and Its Signature

2.1 XML

XML is derived from Standard Generalized Markup Language (SGML). It is much more flexible than HTML, which is also a derivation of SGML. In HTML, only a set of predefined tags can be used, while in XML we can define new tags to customize our needs. Actually, XML is meta-language. For a specified application, we first define XML elements and document structure, namely DTD (Document Type Definitions) and XML-Schema. According to these definitions, we, further, design SAX (Simple API for XML) or DOM (Document Object Model). Then we can store and transmit information by legal XML documents consistent with these definitions. In other words, information is exchanged through such XML documents and interpreted by the SAX or DOM parser interface for the corresponding application.

The mechanism of XML shows that although new tags can be defined flexibly, for a specific application, illegal tags, i.e. those inconsistent with the DTD and XML-Schema of this application, will cause exceptions, or errors, of the SAX or DOM parser. This underlies the verification process of our proposed XML watermarking algorithm, which is discussed specifically in the following section.

2.2 XML Signature

Digital signature is a traditional and widely used method for integrity protection of digital documents. To protect the integrity of XML documents, the W3C issued the XML signature standard [8], a special version of digital signature for XML documents. It not only defines what information in the original XML document and how to be signed, but also defines an XML schema to attach the signature to the original document to generate a signed document. This signature would allow the receiver to verify whether the message has been modified from the original one.

Suppose there is an XML document, say *M*, which need to be signed when it is sent from the sender to the receiver. According to the secure hash algorithm (SHA-1, for example) and the original XML document, the sender generates a digest, *H(M)*, and encrypt it with the sender's private key, getting $H_s(M)$, which is the signature attached to *M*. As a result, the signed XML document M_s is sent to the receiver. When the receiver receives the message, $\hat{M}_s$, it generates the digest, $H(\hat{M}_s)$, for the information in the received XML document. And the attached signature $H_s(\hat{M})$ in $\hat{M}_s$ is decrypted with the sender's public key, resulting in the sender's digest $H(\hat{M})$. If $H_s(\hat{M})$ and $H(\hat{M})$ are the same, then the receiver knows that M is not modified, otherwise the M is modified. Here in Fig. 1 we give a simple example of detached XML digital signature. Since such signature has to be attached to the original XML document, additional storage and bandwidth are required to keep and transmit the whole signed XML document so that its integrity can be ensured. The proposed XML watermarking scheme in this paper, however, has no such problem.

3. XML Watermarking Scheme

3.1 HTML Watermarking

The early research of watermarks for text documents mainly focused on text images and formatted documents, PDF and WORD, for instance. Because there is no format in such plain text documents as HTML web pages and XML documents, these algorithms cannot be applied to these documents. Later people turned to the characteristics of HTML itself to explore proper watermarking algorithms for HTML documents. The works in that period presented some HTML watermarking algorithms, which are still the base of current HTML watermarking software. The basic idea of them is to embed watermarks into HTML

documents by adding space and tab into them, because such space and tab do not affect the display of HTML web pages [9]. A disadvantage of these methods is that the file size of HTML documents increases after being watermarked. This problem is discussed thoroughly in [14]. To conquer this, Zhao and Lu [14] used another watermark embedding method, called Upper-Lower Coding, or ULC, for HTML documents. It is based on the case-insensitivity of HTML tags. Specifically, watermarks are embedded into HTML documents by altering the case of letters in HTML tags, which does not affect the display of HTML web pages too. Fig. 2 illustrates the increase in file size of watermarked web pages with STC and ULC, respectively. From the diagrams we can find that file size is expanded greatly when STC is applied while ULC does not increase file size at all.

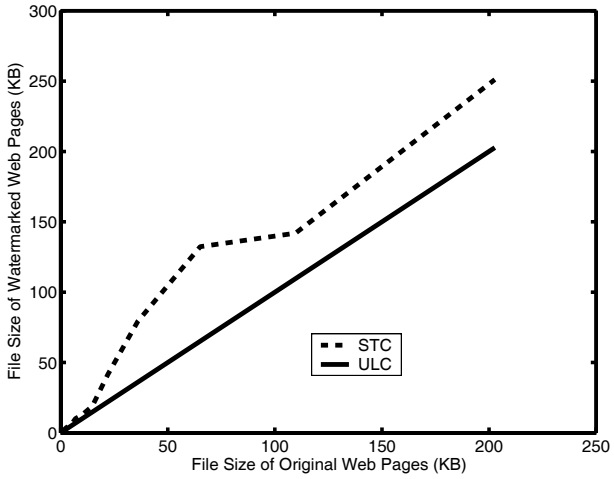

Figure 2: STC's and ULC's effect to file size

Due to such merit of ULC, the XML watermarking scheme proposed in this paper also takes this method. The difficulty is that XML tags, unlike HTML tags, are case-sensitive [1]. Thus the algorithm based on ULC can not apply to XML watermarking directly. However, taking the execution mechanism of XML into consideration and making use of the Exclusive OR (XOR) operation, we propose an effective watermarking scheme for XML documents.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate technique aiming at using a smaller set of uncorrelated variables to represent a number of related variables [15]. This method has been widely used in the realm of pattern recognition as well as some others. In general, the projected vectors of PCA, namely principal components, capture the most expressive features of the original data. Besides, such projection is economy in the sense that the principal vectors, onto which original data are projected, are uncorrelated. Due to these fine properties, PCA has already been used in the watermarking world. In [14], watermarks for HTML web pages are generated with the PCA technique. The results are promising. Similarly, we generate watermarks for XML documents also based on PCA. However, here the matrix of an XML document is formed according to the following principle. When constructing the matrix, letters of different cases in XML tags are mapped to a same integer while those outside to different integers. This is because watermarks are embedded in the tags by altering the case of letters in them, thus the watermark generating method itself should be case-insensitive in tags although XML tags are case-sensitive.

3.3 PCA-based Watermark Generating

A matrix H is first formed from the given XML document: each text line in the document gives birth to a row vector in H. That is all letters in the text line are mapped to integers according to their indexes in the code chart adopted by the document, ASCII or UNICODE, for example. However, different strategies are applied to letters in and outside XML tags respectively. Specifically, letters in tags are case insensitive in our coding scheme, whereas those outside are case sensitive. In order to make all rows have same length, we take means of `cyclic filling' to expand the length of every row vector to that of the longest one. In this way, we obtain a $R \times C$ integer matrix H, and from H, we obtain a square matrix D:

$$D = H \times H^t, \tag{1}$$

where t denotes the transpose operation. D is then convoluted with a key K (K itself is an $N \times N$ matrix) to get a $(R+N-1)\times(R+N-1)$ matrix I:

$$I = D \otimes K, \tag{2}$$

Where '$\otimes$' denotes the convolution operation. The role of convolution is to diffuse and amplify the effects of possible modifications.

Define the covariance matrix V of I as

$$V = \sum_{i=1}^{R+N-1} (I_i - \bar{I}_R)^t \times (I_i - \bar{I}_R), \tag{3}$$

Where I_i is the i_{th} row vector of I, and $\bar{I}_R$ is the mean of the row vectors of I, i.e.

$$\bar{I}_R = \frac{1}{R+N-1} \sum_{i=1}^{R+N-1} I_i. \tag{4}$$

After Eigen-Decomposition (ED) on V, we choose the first R eigenvectors, $u_1, u_2, \cdots, u_R$, to generate the feature space S (assuming the corresponding eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R \geq \cdots \geq \lambda_{R+N-1}$):

$$S = span\{u_1, u_2, \cdots, u_R\}. \quad (5)$$

Now we can generate the watermark. Firstly, we get $1 \times R$ row vectors $Z_1, Z_2, \cdots, Z_R$, which are defined as

$$Z_i = (I_i - \bar{I}_R) \cdot [u_1 u_2 \cdots u_R], i = 1, 2, \cdots, R. \quad (6)$$

Secondly, they are converted to the binary form, i.e. a sequence of `0' and `1'. Denote the j_{th} element of Z_i as z_{ij}, $j = 1,2,\cdots,R$. Each digit of z_{ij} is converted to its binary representation β_{ij}; all of them are then linked together to generate a binary sequence $W_i = \beta_{i1}\beta_{i2}\cdots\beta_{iR}$, which was taken as the watermark for the i_{th} text line of the XML document, $i = 1,2,\cdots,R$.

3.4 Watermark Embedding

We embed the watermark into the XML document by altering the case of letters in XML tags (called Upper-Lower Coding, or ULC). Obviously, this will not increase the file size of XML documents, which makes our scheme superior to the signature approach. Specifically, we first acquire a binary sequence T_i^b according to letters in the XML tags in every text line T_i, $i = 1,2,\cdots,R$: an upper case letter generates an `1' while a lower case a `0'. Then, we make the XOR operation between T_i^b and the watermark W_i, and use the obtained result to determine the case of letters in the XML tags of T_i: the j_{th} letter is set to be lower case if the j_{th} element of W_i and that of T_i^b are identical, otherwise upper case. When the number of letters is larger than the length of W_i, the index of the letter is mod by the length of W_i.

3.5 Watermark Verification

To check whether a watermarked XML document has been tampered or not, we first generate a watermark for it as described above. Afterward, the XOR operation is made between the generated watermark and the watermarked XML document as in the embedding process. As a result, we obtain another XML document and we can claim that if the watermarked XML document is integral, then the obtained document should be identical with the original one while a tampered XML document will cause the XML parser to report an error because the result of XOR is an illegal XML document. Thus we can judge the integrity of XML documents effectively. This functions because of the case-insensitivity of the encoding scheme in the XML tags (although XML itself is case sensitive) and the properties of XOR operation. The security of this scheme is ensured by the following facts. When the watermarked document D_w is altered to D_w^F, it is of little possibility for them to have identical watermarks and thus an illegal XML document will be retrieved from D_w^F; moreover, it is impossible or computationally unfeasible for attackers to work out the legal key *K*.

4. Experimental Results

We test the effectiveness of the proposed watermarking scheme first and Fig.3 gives an example of the experimental results. Fig.3(a) is the original XML message, broadcasted by the ACME company's procurement application to ACME suppliers [7]. Fig.3(b) shows the watermarked message and Fig.3(c) is the extracted XML message from it. Apparently, the extracted message is totally the same as the original one. Fig.3(d) through Fig.3(f) show possible attacks. Suppose the message is tampered by altering the product code to ``CA350" and the quantity to ``1,000,000". Fig.3(d) shows the extracted message from the tampered one. Obviously, it is illegal. Fig.3(e) is the one with counterfeit watermarks that are generated from the tampered message and a fake key. Its extracted message is given in Fig.3(f), which is also illegal. These experiments testify the effectiveness of the proposed watermarking scheme in protecting the integrity of XML documents.

Besides its effectiveness, efficiency is also important for the algorithm to be used in real applications. Thus we compare our XML watermarking scheme with the W3C XML signature in terms of the consumed time. Note that we implement both of these two methods with Matlab and all experiments are conducted on the same PC. Therefore the resulted time, given in Table 1, is comparative. The table shows the time spent by generating signature and embedding watermarks. And data in it also illustrate the time consumed by verifying signature and watermarks respectively. These data demonstrate the advantages of our proposed watermarking scheme over signature methods in terms of time consumed. The HASH algorithm used in the experiments is SHA-1 [16]. When the document is larger than 20 KB, the time consumed by the watermarking scheme is still about 5 times less than that consumed by signature methods. In fact, for a real application, it is not necessary to retrieve the whole XML document before parsing it. Instead, the retrieved document can be parsed at the same time of retrieving and the verification is stopped when the first mismatch is found. However, in our experiments, the whole XML documents are retrieved first and then compared with the original ones so that we can find out whether the watermarks are correct and the documents are integral. Thus it is expected to use less time to find the tampered XML documents in real applications.

Furthermore, as a watermarking scheme, the proposed algorithm does not increase the file size of XML

documents and the resulted watermarked XML documents are rather simple compared with the signed ones, which expand the file size because of the attached signature, referring to Fig. 1 and Fig. 3. Such advantage makes much more sense for those XML documents smaller than 1 KB, for example business messages transmitted in most B2B and B2C applications.

```
<proc:QuotationRequest xmlns:proc="http://www.acme.com/Procurement">
 <proc:ProductType code='AC350'/>
 <proc:quantity>100,000</proc:quantity>
</proc:QuotationRequest>
```

(a)

```
<PrOc:qUoTaTIOnreQUEsT XMLnS:PROc="HTTp://WWw.aCME.CoM/prOcUREmEnT">
 <PRoC:pRoDUCTTYPE cODE='ac350'/>
 <pROc:QUAnTItY>100,000</PrOC:QUANtITy>
</PRoc:QUOTatioNrEquesT>
```

(b)

```
<proc:QuotationRequest xmlns:proc="http://www.acme.com/Procurement">
 <proc:ProductType code='AC350'/>
 <proc:quantity>100,000</proc:quantity>
</proc:QuotationRequest>
```

(c)

```
<PRoC:QuOTAtiONREQueST xmLNs:ProC="htTP://wwW.AcmE.cOM/PROCurEMeNT">
 <prOc:PrOducttype Code='ca350'/>
 <Proc:qUaNtITy>1,000,000</pROc:qUantitY>
</prOC:quotATIOnReQUESt>
```

(d)

```
<PROC:quOTATionRequESt xMlNS:Proc="HtTP://WwW.AcMe.cOM/pRocurEMENt">
 <pROc:prODuCttYpE CoDe='Ca350'/>
 <pROc:quAntitY>1,000,000</prOc:QUANTity>
</pROC:quOTaTIonreqUEst>
```

(e)

```
<Proc:QUoTatIoNrEqUest Xmlns:PROC="hTTp://wWw.aCme.CoM/ProCUREment">
 <ProC:PRodUcTTyPe cOdE='cA350'/>
 <Proc:QuaNTiTy>1,000,000</PROC:qUanTITY>
</Proc:QUotAtiONREQueST>
```

(f)

Figure 3: Effectiveness test. (a) Original XML message. (b) Watermarked XML message. (c) Extracted message from un-tampered document `b'. (d) Extracted message from tampered document. (e) Tampered XML message with counterfeit watermarks. (f) Extracted message from `e'.

Table 1: Watermark vs. Signature in terms of time consumed

File Size (byte)	223	308	433	757	1300	2170	2390	4100	5440	7890	9480
Embed Watermark (ms)	80	90	93	125	200	375	407	520	609	813	937
Attach Signature (ms)	3870	5100	6620	7730	17910	29900	30480	62340	80000	130180	173090
Verify Watermark (ms)	51	57	78	94	142	359	422	521	594	797	1020
Verify Signature (ms)	3850	5090	6510	7670	17810	29860	30550	62280	79920	129650	171940

5. Conclusions

These extensive experiments lead to the following conclusions. (1) The proposed watermarking scheme is effective in protecting the integrity of XML documents. (2) The watermarking scheme does not increase the file size of XML documents, while signature methods attach the additive signatures to the original documents and thus expand their size. (3) With PCA and ULC, the proposed XML watermarking scheme also outperforms the traditional signature methods in time spent by the algorithms. All these demonstrate that the XML watermarking scheme is a promising tool for the integrity protection of XML documents.

In this paper, we use the convolution operation in the proposed XML watermarking scheme. Although it can diffuse and amplify possible modifications, it is the most time-consuming part of the whole algorithm. To find other simpler techniques with the same function as convolution is one of our future researches. Additionally, to theoretically analyze the security of the proposed watermarking scheme is meaningful for promoting it further to real applications. Another interesting problem is

how much watermark information can be generated for and embedded into XML documents based on PCA and ULC.

Acknowledgment

This work is supported by NSFC under project No. 60573033.

References

[1] Elliotte Rusty Harold, ``XML: Extensible Markup Language", IDG Books Worldwide, Inc., 1999.
[2] Ben Galbraith, Whitnet Hankison et al., ``Professional Web Services Security", Wrox Press Ltd., 2002.
[3] Kevin Dick, ``XML: A Manager's Guide, Second Edition", Pearson Education, Inc., 2003.
[4] T.Bray et al., ed., ``Extensible Markup Language (XML) 1.0, Third Edition", World Wide Web Consortium (W3C), February 2004. http://www.w3.org/TR/2004/REC-xml-20040204.
[5] Blake Dournaee, ``XML Security", McGraw-Hill Companies, Inc., 2002.
[6] B. Atkinson, et al., ``Web services security (ws-security)", http://msdn.microsoft.com/ws/2002/04/Security
[7] Ernesto Damiani, Sabrina De Capitani di Vimercati, Pierangela Samarati, ``Towards securing XML Web services", Proceedings of the 2002 ACM workshop on XML security, pp. 90-96, November 2002.
[8] M. Bartel, J. Boyer, B. Fox, B. LaMacchia and E. Simon, ``XML-Signature Syntax and Processing", World Wide Web Consortium (W3C), http://www.w3.org/TR/xmldsig-core.
[9] S. Katzenbeisser, A. P. Petitcolas, ``Information Hiding Techniques for Steganography and Digital Watermarking", Boston, Artech House, 2000.
[10] Guo-Rui Feng, Lingge Jiang and Chen He, ``Orthogonal transformation to enhance the security of the still image watermarking system", IEICE Transactions on Fundamentals, vol.E87-A, no.4, pp. 949-951, April 2004.
[11] Akiomi Kunisa, ``Digital Watermarking Based on Guided Scambling and Its Robustness Evaluation to JPEG Compression", IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol. E86-A, no. 9, pp. 2366-2375, September 2003.
[12] Minoru Kuribayashi, Hatsukazu Tanaka, ``Video Watermarking of Which Embedded Information Depends on the Distance between Two Signal Positions", IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol. E86-A, no. 12, pp. 3267-3275, December 2003.
[13] Ching-Te Wang, Tung-Shou Chen, and Zhen-Ming Xu, ``A Robust Watermarking System Based on the Properties of Low Frequency in Perceptual Audio Coding", IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol. E87-A, no. 8, pp. 2152-2159, August 2004.
[14] Qijun Zhao, Hongtao Lu, ``A PCA-based Watermarking Scheme for Tamper-proof of Web Pages", Pattern Recognition, vol. 38, pp. 1321-1323, 2005.
[15] I. T. Jolliffe, ``Principal Component Analysis (Second Edition)", Springer-Verlag, New York, Inc., 2002.
[16] http://www.itl.nist.gov/fipspubs/fip180-1.htm.