Design and Implementation of Spatial Data Watermarking Service System

WU Baiyan1, WANG Wei1, PENG Zhiyong2, DU Daosheng1, WANG Chuanjian2
1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
2. International School of Software, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
© Wuhan University and Springer-Verlag Berlin Heidelberg 2010

Abstract  Nowadays, more and more digitalized spatial data are sold and transmitted on the Internet. Thus, there arises an important issue about copyright protection of the digital data. To solve this problem, this paper has designed and implemented a spatial data watermarking service (SDWS) system which can provide a secure framework for data transaction and transfer via the Internet and protect the rights of both copyright owners and consumers at the same time.

Keywords  spatial data transaction; digital watermarking; watermarking protocol; mobile agent; digital signature

CLC number  P208

Introduction

The applications of GIS have walked into our everyday life. As the main contents of GIS, the spatial data has great value. The transactions and transfers of digitalized spatial data on the Internet have become more common than ever before. Unfortunately, the unauthorized duplication and distribution of these digital data is unbelievably quite easy under the network environment. This creates urgent demands for a secure and fair data trading environment that can provide effective protection mechanisms during the data transaction on the Internet. Establishing such kind of environment needs some key supporting technologies, such as digital watermarking, watermarking protocols, and so on.

The digital watermarking techniques are considered as one of the most promising solutions for copyright protection. A digital watermark can be characterized as additional signals added to certain digital contents imperceptibly or perceptibly and can be detected sometime later to assert the lawful ownership of the original digital contents. Since the very first introduction of its concept, a wide variety of digital watermarking schemes have been proposed aiming at protecting the copyright of spatial data such as remote sensing images and vector data.

However, the digital watermarking techniques alone cannot terminate the abuse of spatial data because a buyer whose watermark has been detected in unauthorized copies can claim that the unauthorized copies originated from the seller but not from him! To deal with this kind of problem, a number of watermarking protocols [3-6] have been proposed to ensure that the buyers have no way to deny the data abuse and the sellers also have no way to frame the buyers.

Using the digital watermarking techniques and wa-
termarking protocols, we have designed a spatial data watermarking service (SDWS) system that can provide a secure and fair trading environment for spatial data. In this paper, we will explain the watermarking algorithm\[1-2\] used in the SDWS briefly and focus on the system architecture, the database design, the spatial data trading procedure and the implementation of SDWS.

1 Watermarking algorithms of SDWS

1.1 Framework of the algorithm

The framework of the algorithm is implemented based on Fig. 1. During the watermark insertion procedure, the mechanism works as follows: ① Define the metric measure for the disjoint spatial relations; ② Employ the Hilbert Curve to divide the map layer into different data patches and then extract the cover data; ③ Hash each cover data using the primary key concatenating with the most significant bits of the cover data. Then assign the cover data to a subset according to the value produced by the hash value modulo $m$. $m$ is the number of the subsets. ④ Generate the BCH code to be embedded using the original watermark and then insert one bit of the processed watermark into each subset. The watermark bits are repeatedly embedded.

The watermark detection procedure reads as follows: ① Divide the map layer into data patches as the watermark insertion procedure does; ② Compute the cover data; ③ Use the same primary key as the watermark insertion used to hash the cover data and build the data subsets; ④ For each data subset, detect one watermark bit; ⑤ Compare the detected watermark with the original watermark and make a judgment according to the matching level.

1.2 Definition of the metric measure for disjoint spatial relations

The method proposed in this paper can hide the watermark bits sequence into the disjoint topological relations of map objects. To describe the degree of the disjoint relations, a metric measure is defined. For example, there are two disjoint polygon objects $A$ and $B$. If $A$ is the reference object, then the metric measure is defined as follows:

$$M^j_B = \text{area}(B) / \text{area}(B \oplus \text{dist}(\partial B, \partial A))$$  \hspace{1cm} (1)

1.3 Cover data extraction

Given the original polygon map layer $L_0$, after the data patches are built, all the data patches $P=\{p_1, p_2, \cdots\}$ are formed. $\forall p_i \in P$, suppose there are $\kappa$ features in $p_i$, denoted as $f^i_1, \cdots, f^i_j, \cdots, f^i_{\kappa}$. $\forall f^i_j \in p_i$, for each $t \in \{1, 2, \cdots, \kappa\}$ and $t \neq j$, if $f^i_t$ is apart from $f^i_j$, then we consider $f^i_j$ as the reference object and get the metric measure $(M^t_j)'$ according to the Eq. (1). Finally, we can get metric measures $M' = \{ (M^1_j)', (M^2_j)', \cdots \}$. For $M'$, we first use the statistical method Box Plot to clean the outliers and then compute the values of mean, median, the first quartile $Q_1$ and the third quartile $Q_3$. Finally, we get the value $C^j$ according to the following formulation:

$$C^j = (\text{mean} - \text{median})/(Q_3 - Q_1)$$  \hspace{1cm} (2)

When the above data process is completed, we can
get the data set \( C = \{C_1, C_2, \ldots, C_i, C_{i+1}, \ldots\}\). \( C \) will be used as the cover data for watermarking.

### 1.4 One watermark bit insertion

We first divide the cover data \( C \) into \( m \) subsets denoted as \( D_1, \ldots, D_i, \ldots, D_m \) by a cryptographic keyed hash function. \( m \) equals to the product of the length of the watermark bit sequence and the embedding times of the watermark bit sequence. For each of the to-be-watermarked secret (keyed) subsets \( D_i \), one watermark bit is embedded. The solution is as follows.

1) In a data axis, define a reference point \( O \). Separate \( O \)'s left part and right part of the data axis into lines with uniform length \( d \). Number the lines as Fig. 2(a) shows and map the elements of \( D_i \) onto the data axis.

2) When embedding the watermark bit 1 into \( D_i \), adjust the coordinates of the related polygons under some constrains to make the cover data that fall into the lines whose numbers are even move into the nearest lines whose numbers are odd, as Fig. 2(b) shows.

3) When embedding the watermark bit 0 into \( D_i \), adjust the coordinates of the related polygons under some constrains to make the cover data that fall into the lines whose numbers are odd move into the nearest lines whose numbers are even, as Fig. 2(c) shows.

The constraints for adjusting the coordinates include:  
① the coordinate distortions induced by watermarking should not exceed the map precision tolerance;  
② the watermarking process should not change the types of the topological relations between objects. For example, the original disjoint spatial relations should not be changed into the touching or overlapping or any other types of relations during the watermarking process;  
③ the relative relations of metric measures also should not be changed by watermarking;  
④ the shape distortions of the polygons should not be too big. As in the watermarking process, we always have the global view of the objects, we can easily meet the latter three needs. The value of the step length \( d \) is decided by the map precision tolerance.

### 1.5 Watermark detection

Given the possible tampered polygon map layer \( L_w \), first \( L_w \) is separated into data patches. Then cover data is extracted and divided into \( m \) subsets. For each subset \( D_i \), its elements are mapped onto the same data axis as used in the watermarking process. Let \( n_e \) denote the number of the elements that fall into the evenly numbered lines in the data axis and \( n_o \) denotes the number of the elements that fall into the oddly numbered lines. If \( n_e < n_o \), then watermark bit 1 is extracted. If \( n_e > n_o \), then bit 0 is extracted. If \( n_e = n_o \), no bit is extracted. Because the watermark bit sequence is embedded several times, a majority voting schema is employed to decide what the watermark bit sequence is.

### 2 Spatial data watermarking service system

#### 2.1 System architecture of SDWS

The system architecture employing the B/S framework is shown in Fig.1. Three roles are involved in this system, namely the seller, the buyer and the trusted third party (TTP). The seller and the buyer are the terminal users of the system. In the server of the TTP, the certification authority (CA) center and watermarking center are configured. A user identified by the digital certificate issued to him or her by the CA center communicates with the watermarking center through a trusted channel.

#### 2.2 Design of database in the SDWS system

There are three tables in the database of the SDWS system, namely “UsersTab”, “ProductsTab” and “TransactionsTab”. “UsersTab” records the information of registered users including users’ identification...
ID, e-mail address, certificate that contains the public key and so on. “ProductsTab” records the basic information of the spatial data products that have been registered in the SDWS system. “TransactionsTab” records the information of spatial data transactions that have been completed successfully. The three tables and the relationship among them marked by correlation multi-lines are shown in Fig. 3.

2.3 Spatial data trading procedure in the SDWS system

The spatial data trading procedure is shown in Fig. 3. The whole procedure can be divided into the following steps.

1) The CA center issues a key pair including a private key and a public key to the TTP denoted as $T$, the seller denoted as $S$ and the buyer denoted as $B$, respectively. The key pair for $T$ is represented as $(PK_T, SK_T)$. $PK_T$ is a public key and $SK_T$ is a private key. Similarly, the key pair for $S$ is $(PK_S, SK_S)$ and for $B$ is $(PK_B, SK_B)$.

2) Users including sellers and buyers submit their basic information and public key certificate to $T$ to be registered into the “UsersTab”.

3) Sellers log in the SDWS system and then submit the information of their database product denoted as $X$, including the database owner name, the database schema and the metadata. All the information is saved in the “ProductTab”.

4) After signing a buy-sell contract, the buyer starts a transaction by sending the $ID_X$ denoting the ID of the database product that he or she wants to buy and the $ID_S$ denoting the ID of the product owner to $T$.

5) $T$ generates a transaction number denoted as $N$. $N$ represents this transaction between $S$ and $B$. $T$ sends $N$, $ID_X$, $ID_S$, $ID_B$ and the signed information to $S$ and $B$ respectively and asks them to check this transaction.

6) The seller $S$ and the buyer $B$ check the signed information to make sure that the received message is from $T$ and if it is, confirm the transaction $N$. Then $S$ sends $(N, ID_S, W_S)$ and the signed information $\text{SIGN}_S(N, ID_S, W_S)$ to $T$ and $B$ sends $(N, ID_B, W_B)$ and the signed information $\text{SIGN}_B(N, ID_B, W_B)$ to $T$.

7) $T$ checks the signed information of $S$ and $B$. If the two checking results are both true, then $T$ generates watermark $W_T$. The final watermark $WM$ to be embedded into the database product consists of $W_T$, $W_S$, $W_B$, $N$ and the timestamp.

8) The seller $S$ applies for watermarking service and submits the database address, user name and password to $T$.

9) $T$ dispatches a mobile agent named watermarking agent to $S$’s database host to watermark the database product and encrypts the watermarked database by public key of $B$. After watermarking and encrypting the database, the agent sends the process result to $T$. If the process is completed successfully, $T$ informs $S$ that the copy of the database product can be transferred to $B$ now.

10) After the transaction is completed, the information of this transaction is saved into “TransactionsTab”.

2.4 Sub-procedure of users register

In order to sell or buy spatial data in the SDWS system, users must first complete the user register process. The information users must submit in the register process include e-mail address and public key certificate. If the register processes are completed successfully, the SDWS system will inform users by e-mails. The public key certificate will be used to check a user’s identity. The user register procedure is shown in Fig. 4.

2.5 Sub-procedure of data products register

When the user registration process is successfully completed, users can then register their data products in the SDWS system. The procedure for the data products registration is shown in Fig. 5.
The administrator’s review process in the above procedure shown by Fig.5 is very important for protecting seller S’s benefits, because it keeps a buyer B from registering the data product bought from S in the SDWS system and thus protects S’s benefits. Fig.8 shows the administrator’s review procedure.
3 Implementation of prototype system

3.1 Development environment

A prototype of the spatial data watermarking service system is developed. The development language is JAVA and the development tools include eclipse, weblogic, Aglets and so on. Aglets is a mobile agent technology and was developed using JAVA by IBM company\(^5\) in Japan. Fig.7 shows the communication in the SDWS system.

We employ the mobile agent technology to embed and detect the watermarks in the spatial database products. The mobile agents do not need the continuing connected network. The watermark detection can be carried out during the casual time of the remote host and the detection results can be sent back asynchronously. Here we use the Master-Slave mode which consists of TestMaster agent, Add_TestSlave agent and Detect_TestSlave agent.

TestMaster is responsible for generating sub-agents and dispatching them to remote hosts to do some specific work, communicating with the remote hosts and destroying the dispatched sub-agents. Fig.10 is the flowchart of TestMaster.

![Diagram of TestMaster](image-url)
The Add_TestSlave agent, which is generated by TestMaster, is responsible for watermarking the databases in remote hosts. Add_TestSlave is dispatched to and runs on the remote hosts. It connects to the database to watermark the data products and reports the watermarking results to TestMaster.

The Detect_TestSlave agent, which is also generated by TestMaster, is responsible for detecting the watermarks hidden in the databases in remote hosts. Similar to Add_TestSlave, Detect_TestSlave also runs on remote hosts. It detects the watermarks and reports the detection results to TestMaster.

3.2 Watermarking experiments

The data used for the experiment are part of a residential area map. The tolerance of the coordinates is 1 m. In the watermarking process, the adjusting amplitude is 0.5 m. Fig.11 shows the good invisibility of the schema.

The original watermark bit sequence is 110. The bit sequence is repeatedly embedded 18 times. Table 1 is the majority voting results.

4 Discussion and conclusion

This paper has designed a spatial data watermarking service system and presented a prototype of the SDWS system. The system owns the following characteristics to make sure that the spatial data transactions in the SDWS system are secure and fair to both sellers and buyers.
Table 1  Data cropping tests

| Cropped quantity(%) | The original watermark | Cropped map | The detected watermark | Matching level (%) |
|---------------------|------------------------|-------------|------------------------|-------------------|
| 30                  | 01110111010001001111001(why) | ![Cropped map 30%](image) | 01110111010001001111001 | 100               |
| 50                  |                        | ![Cropped map 50%](image) | 01110111010001001111001 | 100               |
| 70                  |                        | ![Cropped map 70%](image) | 111111111111111001111011 | 71                |

(1) The watermarking method is based on the objects instead of vertices and this brings advantages. ① The capability of the method has nothing to do with the number of the map vertices. It is only decided by the number of the features. ② In the watermarking process, the global view of objects are always held to make sure that the shape distortions are preserved and no topology conflict is produced.

(2) The watermark sequence is embedded in the disjoint relations, not directly in the vertices, so it is very robust to noise attacks that most existing algorithms cannot counter. Besides, as the cover data is invariant to map translation, zooming and rotation, the proposed schema is inherently robust to these kinds of attacks. In addition, it is also robust to vertex attacks such as simplification and interpolation.

(3) In the user registry process, users need to submit their public key certificates which are used to authenticate the users’ identification so that the users cannot deny any activities in the SDWS system.

(4) In the registry process of the data product, the administrator’s review procedure stops the watermarked data products being registered in the SDWS system again. That means, B cannot sell the data product bought from S in the SDWS system so that S’s benefits are protected.

(5) When finding possibly tampered data product X, S asks T to detect the watermark in X. First, T looks
for all the transaction records of $X$ in the transaction table and fetches the watermarks $WM_1$, $WM_2$, $\ldots$, $WM_n$ and the corresponding keys, i.e., $Key_1$, $Key_2$, $\ldots$, $Key_n$ from the matched records. Then $T$ uses $Key_i$ to extract the watermark in $X$. For $Key_i$, if $WM_i$ is detected, then $X$ is proved to be a pirated copy. Furthermore, it can be easily proved that the pirated copy originates from the buyer $B$ not the seller $S$. Because only $B$ can decrypt the data product, which is encrypted by $B$’s public key. $B$’s identity can be known from the matched transaction record.

(6) In order to keep the data integrity and security during the migrating process, the important watermarking parameters contained in the aglets are encrypted and transmitted using SSL protocol. When in the running state, the aglets are sandboxes to a certain degree to ensure the running security.

This paper proposes a possible and feasible solution for building a secure and fair spatial data transaction framework, and a prototype system is implemented.

References

[1] Wu Baiyan, Wang Wei, Miao Dandan (2008) 2D vector map watermarking based on the spatial relations [C]. Proceeding of International Conference on Earth Observation Data Processing and Analysis, Wuhan

[2] Wu Baiyan, Wang Wei, Peng Zhiyong, et al. (2009) A new algorithm for watermarking building polygons [C]. International Conference on Digital Image Processing (ICDIP 2009), Bangkok, Thailand

[3] Nasir Memon, Wong P W (2001) A buyer-seller watermarking protocol [J]. IEEE Transactions on Image Processing, 10(4)

[4] Chang Chin-Chen, Chung Chi-Yien (2003) An enhanced buyer-seller watermarking protocol [C]. Proceedings of ICCT2003, Beijing

[5] Lei Chin-Laung, Yu Pei-Ling, Tsai Pan-Lung, et al. (2004) An efficient and anonymous buyer-seller watermarking protocol[J]. IEEE Transaction on Image Processing, 13(12)

[6] Chen Xiaosu, Zhu Dali, Liu Jian (2007) A practical digital watermarking protocol based on PKI-CA [C]. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Thailand

[7] IBM Corp (1998) The IBM Aglets workbench [OL]. http://www.trl.ibm.co.jp/aglets.html

[8] Danly B Lange (1998) Mobile agent with Java: the aglet API [J]. World Wide Web Journal, 3

[9] Lang D, Mitsuru O (1998) Programming and deploying java mobile agents with aglets[M]. New York: Addison-Wesley Longman Publishing Co., Inc