Manuscript

Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Authenticating topological integrity of process plant models through digital watermarking

Zhiyong Su · Lang Zhou · Jianshou Kong · Yuewei Dai

Received: date / Accepted: date

Abstract Process plant models which feature their intrinsical complex topological relation are important industrial art work in the field of Computer-Aided Design (CAD). Compared with the widely studied watermarking based geometrical information protection and authentication techniques for traditional mechanical CAD drawings, topology authentication is still in its infancy and o☐ers very interesting potentials for improvements. This paper investigates the topology authentication problem for process plant models. We propose a semi-fragile watermarking based algorithm to address this interesting issue. We encode the topological relation among joint plant components into the watermark bits based on the hamming code. A subset of the model's connection points are selected as mark points for watermark embedding. Then those topology sensitive watermark bits are embedded into the selected mark points via bit substitution. Theoretical analysis and experimental results demonstrate that our approach yields a strong ability in detecting and locating malicious

Zhiyong Su
School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
Tel.: +8625 84315467
Fax: +8625 84317332
E-mail: suzhiyong@njust.edu.cn

Lang Zhou
College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210046, China
E-mail: yzzhoulang@126.com

Jianshou Kong
School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
E-mail: kongjs77@163.com

Yuewei Dai
School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
E-mail: daiywei@163.com

topology attacks while achieves robustness against various non-malicious attacks.

Keywords Watermarking · Semi-fragile watermarking · Topology authentication · Process plant model · CAD

1 Introduction

Today's process industries are global and characterized by complex design and engineering. This calls for collaborative product development among designers, manufacturers, suppliers, etc. The Computer-Aided Plant Design system is now increasingly used in process industries for helping increase productivity and collaboration to meet the challenges of complex plant design projects. Collaborative design is the process where multidisciplinary designers participate in design decision-making and share product information across enterprize boundaries. During collaboration, a manufacturer may share process plant models, as one kind of 3D CAD (Computer-Aided Design) models, with its supplier as design specifications. They may also share process plant models with their customers for analysis and simulation purposes. Therefore, particular attention to integrity authentication is necessary to companies when sharing process plant models with their suppliers or customers.

The complex topological relation is one of the most important items to be authenticated in process plant models. Computer-Aided Plant Design systems mainly focus on optimizing the plant layout while the traditional mechanical CAD industry mainly concentrates on the geometrical modeling [1]. Plant layout aims to find the most economical spatial arrangement of process vessels, equipments and their interconnecting pipes which satisfies construction, operation, maintenance, and safety requirements [8]. This is an important aspect in the design of process plants since a good layout will ensure that the plant functions correctly and will provide an economically acceptable balance between the many, often conflicting, design constraints [7]. Moreover, various construction documents, such as isometrics, orthographics, etc., are automatically generated from the process plant model on the basis of complex topological relation among plant components.

The problem of topology authentication for process plant models can be classified into the following two aspects: joint plant components authentication and joint ends authentication. Joint plant components authentication aims to make sure that whether the joint plant components of each plant component are changed or not. Furthermore, joint ends authentication verifies whether the exact joint ends between the two joint plant components are modified or not. That is to say that, for each plant component, the problem of topology authentication targets to verify not only its joint components, but also the exact joint ends, since a plant component usually has more than one joint ends.

Digital watermarking provides an effective and reasonable solution for the integrity authentication of multimedia objects [27]. It has been widely studied and used for authenticating or protecting multimedia objects including sound [26], still image [3], video [18], three-dimensional(3D) models [28, 6, 17], etc. Process plant models, as one kind of 3D CAD models, can also be regarded as a full-fledged multimedia data type, although this may not be a common perception [21]. However, relatively few watermarking algorithms have been proposed for 3D CAD models especially process plant models. Furthermore, methods to watermark geometrical information have been the focus of the research in watermarking CAD models including CAD-based drawings, parameterized curves and surfaces, etc. For CAD-based drawings, Peng et al. proposed two watermarking scheme for 2D CAD engineering graphics by modifying coordinates of vertices based on improved difference expansion and log-polar transformation respectively [24, 25]. Lee et al. presented a robust watermarking scheme based on geometric features with k-means++ clustering for 3D CAD drawings [16]. The proposed scheme embeds the watermark into the geometric distribution of POLYLINE, 3DFACE, and ARC objects in main layers. Kwon et al. described two algorithms for 3D CAD drawings by selecting LINE, FACE, and ARC components as watermark carriers [11, 10]. For parameterized curves and surfaces, Ohbuchi et al. presented a watermarking scheme for 3D NURBS curves using reparameterization [22]. Lee et al. proposed a method for watermarking NURBS data using two-dimensional virtual images [15]. A robust non-blind watermarking scheme for subdivision surfaces was presented by Lavoué [14]. They embed the watermarks into the frequency domain by modulating spectral coefficients of the subdivision control mesh. Kwon et al. presented a blind watermarking scheme for rational Béier and B-spline curves and surfaces. Their algorithm is shape-preserving and robust against the affine transformations and Möius reparameterization which are commonly used in geometric modeling operations in CAD systems [12]. In summary, existing watermarking schemes for CAD models mainly target the geometrical information protection or authentication. Topology authentication for process plant models is still in its infancy and offers very interesting potentials for improvements.

In this paper, we dedicate to tackle the problem of topology authentication for process plant models. And a semi-fragile watermarking scheme is proposed for this interesting issue. The first contribution of this paper is the design of a novel semi-fragile watermarking based scheme for the topology authentication problem. This idea is inspired by existing fragile or semi-fragile watermarking schemes for authenticating the integrity of various multimedia objects. The semi-fragile technique proposed in this paper is vulnerable to even very slight modifications of the topological relation among plant components. Furthermore, it is also capable of locating and identifying the attacked regions. The second contribution of the paper is that we encode the topological relation into the singular watermark bits for each mark connection point. So any attack which ruins the topological relation will result in the modification of extracted watermark bits.

The remainder of the paper is organized as follows. We give a brief introduction of the topological relation of process plant models and review some related techniques in Section 2. After that we describe in detail the procedure of embedding and extracting the watermarks in Section 4. Section 5 demonstrates and discusses the experimental results. Conclusion and future work follow in Section 6.

2 Preliminaries

This section reviews some related techniques used in our scheme. Section 2.1 describes the structure of process plant models with the focus on topological modeling. Section 2.2 and Section 2.3 discuss the logistic map and hamming code approaches used for the watermarking generation, respectively. Finally, Section 2.4 reviews the principal component analysis method employed to produce a□ne invariants for watermark embedding.

2.1 Topological modeling of Process plant models

The process plant model covers three kinds of information: geometrical information, engineering information, and topological information. Geometrical information describes the shape and 3D positions. Engineering information refers to design constraints, engineering disciplines and so on. Topological information provides the complex topological relation among di□erent plant components. We give a detailed introduction of the topological relation representation among various joint plant components in this section.

Process plant models feature their intrinsical complex topological relation rather than their geometrical shape represented by various basic solid entities, such as box, cylinder, prism, sphere and so on. Topological modeling concerns with the most economical spatial arrangement of process vessels, equipments and their interconnections that satisfies construction, operation, maintenance, and safety requirements. And it poses significant limitations on the type, size and location of plant components. Not only should the layout represent the interconnection among joint plant components, but it should also describe their exact interconnection ends. Only the two ends of di□erent plant components which satisfy the specific requirements, such as pipe diameter, end type, pressure rating, and flow direction, can then be connected.

There are mainly two popular ways, which are widely used in many commercial process plant design softwares, to represent the end connection. One is connection points [4], the other is the order of plant components stored in the file. This paper aims to watermark process plant models which describe the end connection by virtue of connection points.

The core structure of the connection point consists of geometrical information, topological constraint, handle value and various engineering properties, which are shown in Fig. 1. The connection point is, in fact, a point entity. And its geometrical information indicates the actual location. The connection point is normally defined as the center point of the end face. Topological constraint covers its joint connection point and the corresponding plant component it subjects to. Each connection point may have one joint connection point at most. The handle value is an abstract reference to an entity in the process plant model. This value (i.e., an identification number) is unique and is not altered even if the entity is modified (i.e., translated, rotated and scaled). Fig. 2 shows connection points of a simple pipeline. Take the connection point $P_{i,1}$ for example, its corresponding plant component is C_i and $P_{i+1,0}$ is its joint connection point.

It is worth mentioning that, in Computer-Aided Plant Design systems, connection points are added, deleted and transformed along with their corresponding plant components. And the maintenance of connection points is carried out automatically by Computer-Aided Plant Design systems without the need of human intervention.

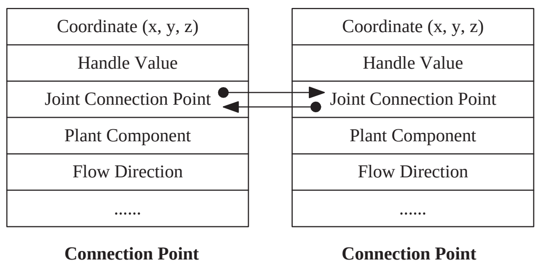

Fig. 1 The core structure of connection points

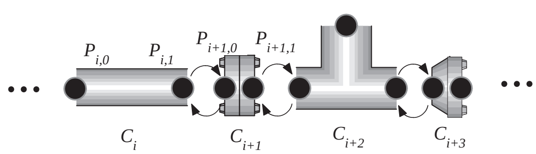

Fig. 2 An example of connection points of a simple pipeline. Note that all the connection points are scaled for better illustration

2.2 Logistic map

The topological relation among joint plant components is involved in the watermark generation using the deterministic logistic map in this paper. Logistic map is a chaotic map that can generate chaotic signal which has the extreme sensitivity to initial conditions, randomness and uniform distribution [20]. Due to these characteristics, it has been widely used for watermarking and encryption [20, 2]. The function used in this paper is defined as:

$$x_{n+1} = ax_n(1 - x_n), \quad (1)$$

where a is the control parameter and x_n is the current value of the mapping in time with an initial value x_0. The sequence iterated with an initial value is chaotic when $a > 3.5699456$. And different sequences will be generated with different initial values.

2.3 Hamming code

The hamming code, first proposed by R.W. Hamming[9], is employed both in the watermark generation and extraction stage of our scheme. Parity check is the basic idea of the Hamming code. The hamming code detects errors by ensuring that each parity check bit and its corresponding data bits achieve the goal of even parity. The number of parity check bits is determined by the hamming inequality rule. One of the most widely used hamming codes is (7, 4), which encodes four data bits (D_1, D_2, D_3, D_4) into seven bits by adding three parity check bits (P_1, P_2, P_3). Fig. 3 depicts the normal permutation form of the seven bits. Fig. 4 shows the creation of parity check bits. The lines indicate the relationships between the data bits and the three parity check bits. In this paper, we utilize the hamming code (15, 11) for generating the content-based watermark bits as well as detecting the tampered plant components and connection points.

P_1	P_2	D_1	P_3	D_2	D_3	D_4

Fig. 3 The usual form of the hamming code (7, 4)

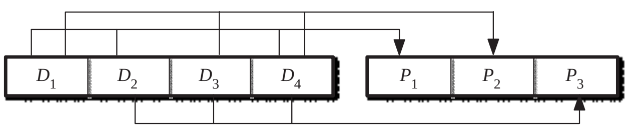

Fig. 4 The way to produce the three parity check bits of the hamming code (7, 4)

2.4 Principal component analysis

PCA is employed to produce the PCA coordinate system and make the system robust against similarity transformation attacks (i.e. translation, rotation, and uniform scaling) in our topology authentication scheme.

PCA was first proposed by Karl Pearson [23] in 1901 and has been widely used in the realm of pattern recognition as well as digital watermarking for multimedia data such as still image, video and three-dimensional (3D) model including 3D geometric CAD model[19, 13, 5]. Let M be a polygonal mesh model with n vertices and let V denotes the sets of vertices of M, respectively. Each vertex $v_i (0 \leq i \leq n-1)$ has three coordinates in the Cartesian space, $v_i = (x_i, y_i, z_i)$. We define the center of the model as v^c by

$$v^c = \frac{1}{n}\sum_{i=0}^{n-1} v_i \tag{2}$$

Where v_i is the ith vertex, v^c is the center of the model. A description of each step of the PCA based transformation used in our scheme is described as follows [5].

The translation invariance is accomplished by translating the model so that its center falls on the center of the coordinate system axes.

$$\hat{v}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i) = v_i - v_i^c = (x_i - x_i^c, y_i - y_i^c, z_i - z_i^c) \tag{3}$$

where $\hat{v}_i$ is the translated vertex and thus we get a new vertex set $\hat{V}$.

The rotation invariance is achieved through rotating the translated vertex $\hat{v}_i$ by

$$\check{v}_i = (\check{x}_i, \check{y}_i, \check{z}_i) = R \cdot \hat{v}_i \tag{4}$$

where $\check{v}_i$ is the rotated vertex, R is a rotation matrix constructed by the covariance matrix C_i. The covariance matrix C_i for each vertex $\hat{v}_i$ is computed in the following way:

$$C_i = \begin{bmatrix} \sum_{i=0}^{n-1}(\hat{x}_i)^2 & \sum_{i=0}^{n-1}\hat{x}_i\hat{y}_i & \sum_{i=0}^{n-1}\hat{x}_i\hat{z}_i \\ \sum_{i=0}^{n-1}\hat{x}_i\hat{y}_i & \sum_{i=0}^{n-1}(\hat{y}_i)^2 & \sum_{i=0}^{n-1}\hat{y}_i\hat{z}_i \\ \sum_{i=0}^{n-1}\hat{x}_i\hat{z}_i & \sum_{i=0}^{n-1}\hat{z}_i\hat{y}_i & \sum_{i=0}^{n-1}(\hat{z}_i)^2 \end{bmatrix} \tag{5}$$

We calculate the eigenvalues of C_i, sort them in decreasing order and compute the corresponding eigenvectors. After normalizing the eigenvectors, we form the rotation matrix R, which has the normalized eigenvectors as rows. Thus we get a new vertex set $\check{V}$ after rotation.

Finally, the uniform scaling invariance is achieved by scaling the set $\check{V}$

$$\acute{v}_i = (\acute{x}_i, \acute{y}_i, \acute{z}_i) = \frac{s_i}{s_{max}}\check{v}_i \tag{6}$$

where

$$s_i = \sqrt{\frac{(\check{x}_i^2 + \check{y}_i^2 + \check{z}_i^2)}{3}}, \tag{7}$$

$$s_{max} = \max(s_1, s_2, ..., s_{n-1}). \tag{8}$$

3 Overview of the algorithm

Our topology authentication scheme consists of two parts: watermark embedding part and watermark extracting part. Fig. 5 shows the overview of our scheme. In the following parts, we call the connection points to be watermarked as mark connection points and the other points as non-mark connection points.

In the watermark embedding part, we first select mark plant components and mark connection points from the model following the mark connection points selecting principle. Then the topological relation among plant components is employed to generate the singular content-based watermark bits for each mark connection point. After that, we embed the topology sensitive watermark bits into each mark connection point by modifying its coordinate according to the watermarks embedding method. Finally we generate the watermarked model.

In the watermark extracting part, the scheme first finds out all mark plant components and mark connection points. Then the tamper detection method is applied to detect and locate the tampered regions and report them visually. In order to identify mark connection points and mark plant components, we extract the watermark bits for each connection point according to the watermarks extraction method. Meanwhile we compute the content-based watermark bits for each connection point through the content-based watermark generation method. After that, the extracted and generated watermark bits are used to label mark connection points and mark components.

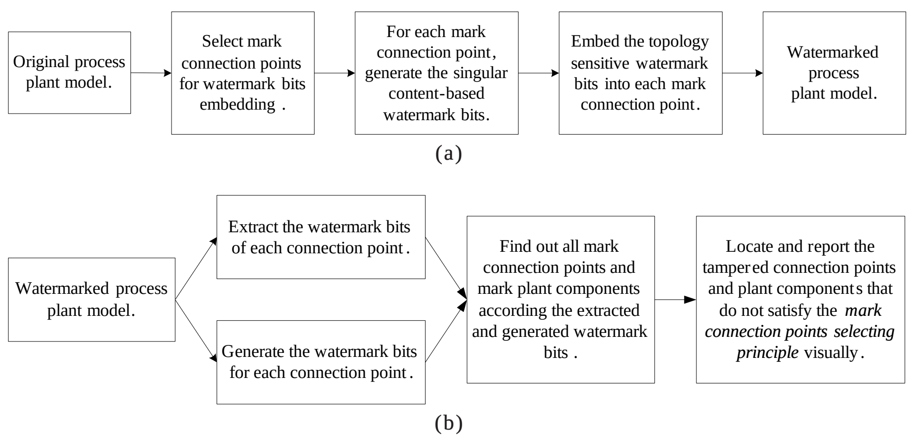

Fig. 5 Overview of our semi-fragile watermarking scheme for topology authentication and verification. (a)Watermark bits embedding; (b)Watermark bits extraction

4 Watermarking based topology authentication

In this section, we discuss our watermarking scheme for topology authentication. We first select a proper portion of connection points from the model for embedding watermark bits. After that we generate content-based watermark bits for each mark connection point. At the end of this section, we describe in detail the procedure of embedding and extracting watermark bits.

4.1 Mark connection points selecting principle

Connection points, rather than geometrical parameters of plant components, are preferred as watermarking targets in this paper. Thus the geometrical shape of the model would not be influenced by the watermark embedding. The principle of mark connection points selecting is described as follows.

First, we select all mark plant components from the model. Initially, all plant components are set as non-mark components. We traverse each pipeline of the model according to the flow direction to get eligible plant components for watermark embedding according to the discipline below.

- One and only one of the two joint plant components must be selected as a mark plant component.
- For a selected mark plant component, there should be no mark components among its 1-ring neighboring components. It means that once a plant component has been chosen as a mark component, its 1-ring neighboring components are no longer eligible.

After the selecting of mark plant components, we set all their connection points as mark connection points for watermark embedding. Fig. 6 illustrates the mark plant components selection of a simple pipeline. From Fig. 6, we can see that the union of selected mark plant components and their 1-ring neighborhood cover all plant components of the model. Therefore, it can be guaranteed that the mark plant components and their mark connection points are uniformly distributed in the model. And this can result in high locating accuracy.

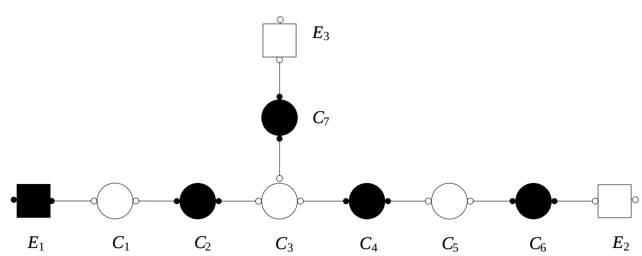

Fig. 6 Illustration of mark plant components selection of a simple pipeline. Circular nodes represent pipe components while rectangular nodes represent equipments. Black nodes are selected mark plant components while white nodes are non-mark plant components

4.2 Content-based watermark generation

We generate singular content-based watermark bits for each mark connection point based on the hamming code (15, 11) and the logistic map method. Handle values of connection points and their corresponding plant components are all involved in the watermark generation.

Assume that a mark plant component C_i with n_i^p mark connection points is connected with a non-mark plant component C_{i+1} with n_{i+1}^p non-mark connection points. Without loss of generality, let $P_{i,j}$ $(j \in [0, n_i^p - 1])$ be a mark connection point of C_i and its joint connection point be $P_{i+1,k}$ $(P_{i+1,k} \in C_{i+1}, k \in [0, n_{i+1}^p - 1])$. Denote the handle values of C_i, C_{i+1}, $P_{i,j}$ and $P_{i+1,k}$ as H_i^c, H_{i+1}^c, $H_{i,j}^p$ and $H_{i+1,k}^p$ respectively. The watermark generation method is described as follows.

1) First, the handel values of the two joint connection points $P_{i,j}$ and $P_{i+1,k}$ are converted into two positive float numbers $F_{i,j}$ and $F_{i+1,k}$ respectively by

$$\begin{cases} F_{i,j} = hash(H_{i,j}^p), \\ F_{i+1,k} = hash(H_{i+1,k}^p), \end{cases} \tag{9}$$

   where hash() is a hash function, $0 < F_{i,j} < 1$ and $0 < F_{i+1,k} < 1$.
2) Then, $F_{i,j}$ and $F_{i+1,k}$ are used as initial values of the logistic function shown in (1). And we perform the logistic function with the two initial values to obtain two float values $L_{i,j}$ and $L_{i+1,k}$ respectively.
3) After that we select 11 bits each from the mantissa parts of both $L_{i,j}$ and $L_{i+1,k}$ under the control of the private key H_i^c and H_{i+1}^c respectively.
4) Let two selected bits be $Bits_{i,j}$ and $Bits_{i+1,k}$ respectively. Then a bitwise XOR operation between the picked mantissa $Bits_{i,j}$ and $Bits_{i+1,k}$ is performed. Finally, four parity check bits, also called the watermark bits $w_{i,j}$, are generated for $P_{i,j}$ from the produced 11 bits data, namely $X_{i,j}$, by the (15,11) hamming code.

It is worth mentioning that there may be some mark connection points with no joint connection points. Given that $P_{i,j}$ is a mark connection point of C_i and it has no joint connection point. Its watermark bits are generated as follows.

1) First, we convert the handel value of the mark connection points $P_{i,j}$ into a positive float number $F_{i,j}$ by

$$F_{i,j} = hash(H_{i,j}^p), \tag{10}$$

   where hash() is a hash function, $0 < F_{i,j} < 1$.
2) Then, the logistic function shown in (1) is performed with the initial value $F_{i,j}$ and consequently a float value $L_{i,j}$ is generated.
3) After that we select 11 bits, denoted as $Bits_{i,j}$, from the mantissa parts of $L_{i,j}$ under the control of the private key H_i^c. Finally, four parity check bits are generated as watermark bits $w_{i,j}$ for $P_{i,j}$ from the produced 11 bits data $Bits_{i,j}$ by the (15,11) hamming code.

Fig. 7 gives an example of how to generate the watermark bits for a mark connection point. These content-based watermark bits are then embedded into mark connection points for topology authentication.

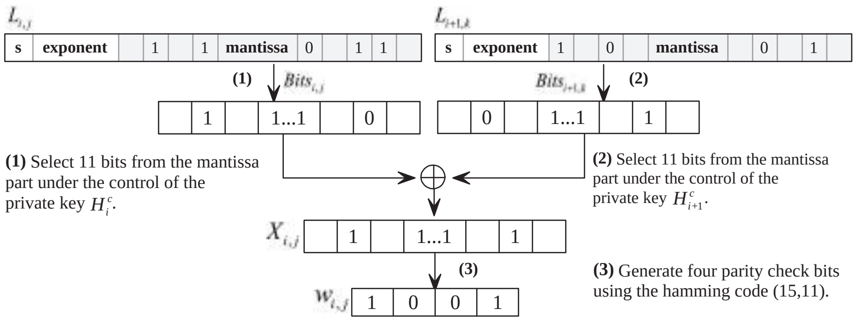

Fig. 7 The example illustrates how the four parity check bits are generated. Data are in the IEEE-754 float32 format

4.3 Watermarks embedding and extraction method

4.3.1 Watermarks embedding

The watermarks embedding method is used to embed the topology sensitive watermark bits into each mark connection point for topology authentication and verification. Provided that $P_{i,j}$ is a mark connection point to be watermarked. Its joint plant component is C_{i+1} with n^p_{i+1} non-mark connection points. Let the total number of joint plant components of C_{i+1} be n^c_{i+1}, which is used as a private key for watermark embedding. The watermark embedding scheme is presented as follows:

1) We first find the sets of neighboring connection points $S(P_{i,j})$ of $P_{i,j}$. $S(P_{i,j})$ is defined as the sets of $P_{i,j}$ and all the connection points $P_{i+1,k}$ of C_{i+1}.

$$S(P_{i,j}) = \{P_{i,j}\} \bigcup \{P_{i+1,k} | P_{i+1,k} \in C_{i+1}, 0 \leq k \leq n^p_{i+1} - 1]\} \quad (11)$$

2) Then the PCA based transformation, described in Section 2.4, is applied to the sets of connection points $S(P_{i,j})$. After that, we convert the transformed point sets to spherical coordinates. Thus $P_{i,j}$ is represented as $(r_{i,j}, \theta_{i,j}, \phi_{i,j})$. This is done in order to achieve robustness against scaling by embedding the watermark bits in the $r_{i,j}$ component of each connection point.
3) For the $r_{i,j}$ component, we select four bits from the mantissa parts of $r_{i,j}$ under the control of the private key n^p_{i+1}, and substitute them with the four bits watermark $w_{i,j}$ generated for $P_{i,j}$.

The watermark embedding process is performed for every mark connection point and finally the watermarked process plant model is archived.

4.3.2 Watermarks extracting

We now discuss how to extract the watermark bits for each connection point from the model. The original model is not needed here. Let C_i be a plant component with n_i^p connection points and n_i^c joint plant components. For each connection point $P_{i,j}$ of C_i, we perform the following parts to extract the watermark bits.

1) First, we find the sets of neighboring connection points $S(P_{i,j})$ of $P_{i,j}$.
2) Then we apply the PCA based transformation to the point sets $S(P_{i,j})$. After that, we convert the transformed point sets to spherical coordinates to get the similarity transformation invariant variable $r_{i,j}$ of $P_{i,j}$.
3) Finally, four bits strings $w'_{i,j}$ are taken from the mantissa parts of $r_{i,j}$ as extracted watermark bits under the control of the private key n_i^c.

4.4 Tamper detection

This procedure is used to detect and locate the tampered plant components and connection ends accurately. Given a watermarked process plant model, we initially set all plant components and their connection points as non-mark plant components and non-mark connection points respectively. The tamper detection and locating procedure is described as follows.

First, we check and find out all of the mark plant components of the model. Let C_i be a plant component with n_i^p connection points.

1) For each connection point $P_{i,j}$ of C_i, we first extract the watermark bits $w'_{i,j}$.
2) Then we compute the watermark bits $w_{i,j}$ for $P_{i,j}$ according to the content-based watermark generation method described in Section 4.2.
3) After that, the watermark bits $w_{i,j}$ is compared with the extracted watermark bits $w'_{i,j}$. $P_{i,j}$ is a mark connection point only if the relation $w_{i,j} == w'_{i,j}$ is satisfied. We label C_i as a mark plant component if it has at least one mark connection point. Otherwise, C_i is set to be a non-mark plant component.

After the labeling of mark plant components and their mark connection points, we detect and locate the tampered regions following the mark connection points selecting principle.

1) For each pipeline of the model, we traverse its plant components according to its flow direction and check if the labeled mark plant components satisfy the mark connection points selecting principle. We set those plant components which do not meet the mark connection points selecting principle as tampered plant components.

2) For each mark plant component, we set it as an unmodified plant component only if all of its connection points are mark connection points. Otherwise, we label its non-mark connection points and their joint plant components as suspicious regions.

5 Performance discussion and experimental results

In this section, we discuss the performance of our semi-fragile watermarking scheme on detecting and locating various attacks and conduct some experiments on a number of process plant models to evaluate the performance of the proposed watermarking scheme. Fig. 8 shows three of the tested models used in our experiments. And their detail information is given in Table 1. The logistic function shown in (1) was seeded with a value a = 4 for 3000 iterations.


(a) (b) (c)

Fig. 8 Three of our tested process plant models used for experiments. (a)Carton board plant; (b)Hydrogenation plant; (c)Styrene plant

Table 1 Lists of three process plant models used in our experiments and their detail information including plant components(PCs), connection points(CPs), mark plant components(MPCs) and mark connection points(MCPs)

Model	PCs	CPs	MPCs	MCPs
Carton board	6810	13964	3365	7002
Hydrogenation	15570	32624	8145	16556
Styrene	18912	38198	9652	19484

5.1 Tamper detection and localization

In this section, we analyze and evaluate the performance of our scheme on detecting and locating the tampered regions on the model. The attacks mentioned in this section include components modification and joint ends modification, which are common operations provided by Computer-Aided Plant Design systems.

5.1.1 Components modification

Topological attacks against plant components mainly cover adding and deleting components.

For plant components adding, there exist two main situations about the joint plant component of the newly added one: non-mark plant component and mark plant component.

- If the newly added plant component is connected with an existing non-mark plant component, it will be labeled as a non-mark plant component during the tamper detection stage. That's because no watermark bits are embedded in its connection points. And this will lead to the mismatch between the extracted and generated watermark bits. Therefore the two joint plant components are all non-mark plant components. Consequently they will be set as tampered plant components since they do not satisfy the mark connection points selecting principle.
- Assume that the newly added plant component C_m is connected with an existing mark plant component C_i. And their two joint connection points are $P_{m,k}$ and $P_{i,j}$ respectively. This kind of attacks changes the topological relation of the mark connection point $P_{i,j}$. Therefore, the extracted watermark bits of $P_{i,j}$ during the watermark extraction stage are different from the watermark bits computed according to the content-based watermark generation method. As a result, the previous mark connection point $P_{i,j}$ will be labeled as a non-mark connection point. And then it, together with the newly added plant component, is set to be tampered.

For plant components deletion, two situations arise: non-mark components deletion and mark component deletion.

- Provided that the non-mark plant component C_{i+1} to be deleted is connected with its joint plant component C_i through their connection points $P_{i+1,k}$ and $P_{i,j}$ respectively. In this case, C_i is a mark plant component and $P_{i,j}$ is one of its mark connection points. There will be no joint connection point for $P_{i,j}$ if the non-mark plant component C_{i+1} is deleted from the model. As a result, the extracted watermark bits of $P_{i,j}$ during the watermark extraction stage are different from the watermark bits computed according to the content-based watermark generation method. Thus the connection point $P_{i,j}$ of the mark component C_i is labeled as a non-mark connection point. Therefore, it is set as a tampered connection point.
- Given that the deleted mark plant component is C_i, which is shown in Fig. 9. This kind of attacks reduces the total number of joint plant components of the non-mark plant component C_{i-1} which is connected with the deleted one. For example, the total number of joint plant components n_i^c of C_i is reduced from 2 to 1 due to the deletion of C_{i+} in Fig. 9. As described in Section 4.3, n_i^c is employed as a key value for both watermark embedding and extraction. Consequently, the modification of n_i^c will lead to the mismatch between the embedded watermark bits and the extracted watermark bits of the mark connection point $P_{i-2,1}$. As a result, the mark

connection point $P_{i-2,1}$ and its joint plant component C_{i-1} are labeled as tampered regions.

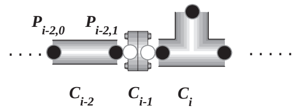

Fig. 9 Illustration of detecting and localizing mark plant components deletion attacks. C_{i-1} is a non-mark plant component. C_{i-2} and C_i are mark plant components. Black points represent mark connection points while white points represent non-mark connection points

Fig. 10 illustrates that our scheme accurately detects and locates the components modification attacks. Fig. 10 (b) and Fig. 10 (c) have been attacked by adding components and deleting components respectively. These regions are labeled as 'A' and 'B' respectively. From Fig. 10 (b) and Fig. 10 (c) we can find that the regions in red are exactly where the tampered operations happen. The experimental results verify the accuracy of our locating procedure.

5.1.2 Joint ends modification

As discussed above, one of the two joint connection points should be a mark connection point. Given that the mark connection point is P while its joint non-mark connection point is P'. The joint connection point of P will be altered if the topological relation between P and P' is modified. Thus, during the watermark extraction stage, the extracted watermark bits will be different from the embedded ones, which are initially generated according to the topological relation between P and P'. Consequently, the two joint connection points and plant components are set as tampered regions.

Fig. 11 illustrates that our scheme accurately detects and locates the joint ends modification attacks. Fig. 11 (b) and Fig. 11 (c) have been attacked by disconnecting the two joint ends geometrically and logically respectively. These regions are labeled as 'A' and 'B' respectively. From Fig. 11 (b) and Fig. 11 (c) we can find that the regions in red are exactly where the tampered operations happen. The experimental results verify the accuracy of our locating procedure.

5.2 Robustness against non-malicious attacks

To evaluated the robustness of our algorithm against various operations provided by Computer-Aided Plant Design systems which can be considered to be non-malicious attacks, we take the watermarked model and apply a combination of rotation, uniform scaling and translation. In our experiment, attack

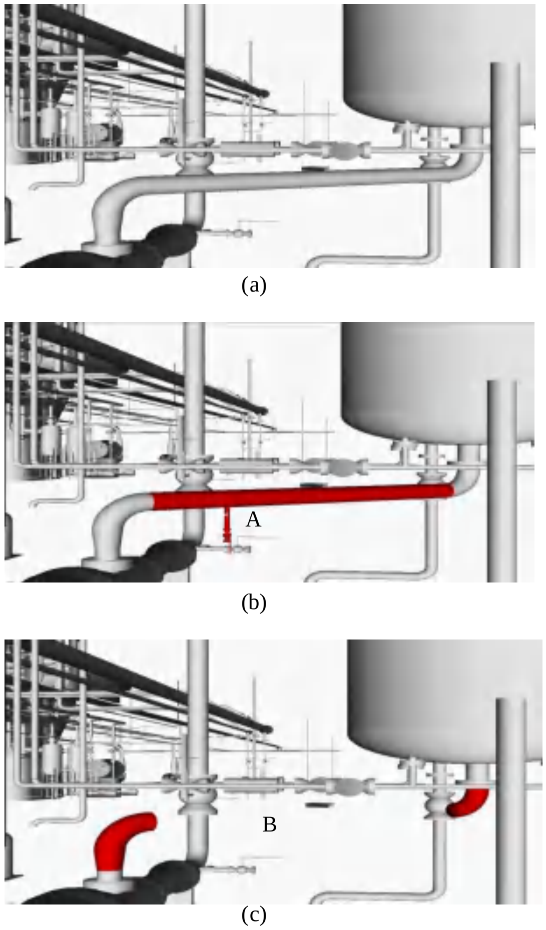

Fig. 10 One example of components modification attacks detecting and locating using our scheme

types are classified as similarity transformation attacks (i.e. translation, rotation, and uniform scaling) and simplification attack. For the robustness, we employ BER (Bit Error Rate) to evaluate the difference between the embedded and extracted watermark bits.

5.2.1 Robustness against similarity transformation

Due to the invariance properties of the PCA based transformation that is applied to the model prior to watermark embedding and detection, the results for similarity transformation attacks are identical to the ones produced when no attack is performed. Therefore, the watermark bits can be extracted without a bit error in spite of the translation, rotation, and uniform scaling.

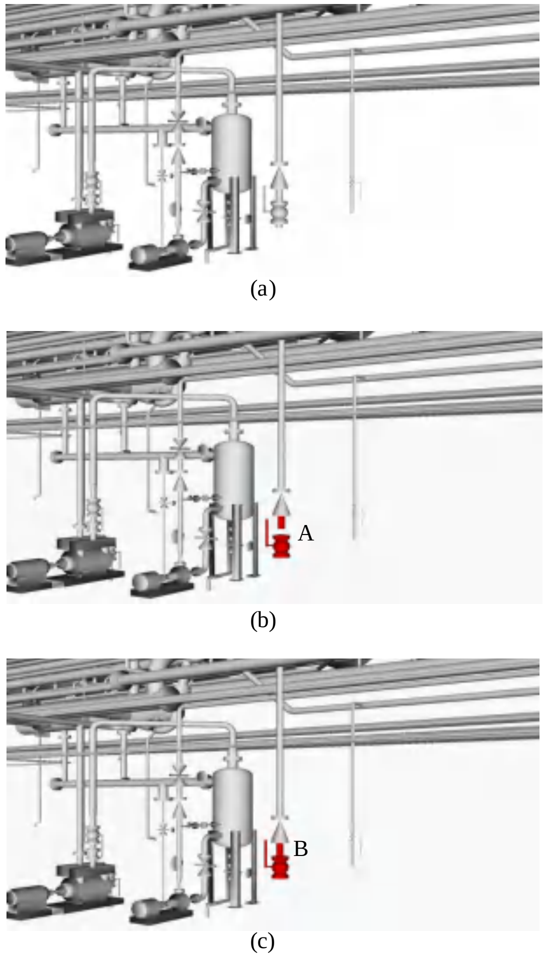

Fig. 11 One example of joint ends modification attacks detection using our scheme

Table 2 BER of the extracted watermark bits in various attacks

Attacks	Carton board	Hydrogenation	Styrene
RST			
Rotation	0	0	0
Uniform scaling	0	0	0
Translation	0	0	0
LOD			
(90% triangles)	0	0	0
(60% triangles)	0	0	0
(30% triangles)	0	0	0

Table 2 presents the robustness evaluation results in terms of the BER under the similarity transformation. The test models are rotated by arbitrary angles, scaled by an arbitrary ratio uniformly, and translated to an arbitrary

position. As seen from the BER values listed in Table 2, our scheme is robust against translation, rotation, and uniform scaling.

5.2.2 Robustness against simplification

Computer-Aided Plant Design systems regularly generate complex models that exceed the interactive visualization capabilities of current graphics systems. The enormous size of process plant models poses a number of challenges in terms of interactive display and manipulation. Several acceleration techniques that reduce the number of rendered polygons have been proposed. Levels of detail (LOD) is one of the key techniques to reduce the model complexity and improve the rendering performance for large scale complex models. It pre-computes different LODs of a given model. At runtime, before rendering each frame, the appropriate LODs to display are selected so that coarser approximations are used for models that are further away or contribute less to the final image.

In this paper, we prefer the connection points rather than the geometrical parameters of plant components as embedding targets. Thus, our embedding method has no influence on the geometrical shape of process plant models and vice versa since LOD can only change the details of entity surfaces. The set of connection points and topological relation among plant components will not be affected. Therefore, our scheme is robust against LOD.

The robustness evaluation results against simplification are also showed in Table 2. We generate three simplified models with different levels for each tested model. From the Table 2 we can conclude that our scheme is invariant to LOD.

5.3 Imperceptibility evaluation

As discussed in Section 5.2.2, our scheme has no influence on the geometrical shape of plant components. Nevertheless, we still give an objective measure for evaluating the quality of a process plant model. The root mean square error (RMSE), as formulated in (12), is used to measure the distortion inflicted on the connection points by our watermarking scheme. It should point out that the geometrical shape of plant components are independent of their connection points

$$RSME = \frac{1}{n}\|P - P'\|, \tag{12}$$

where P and P' are the sets of connection points in the process plant model and its watermarked counterpart, respectively, and n is the number of connection points.

Table 3 details the RSMEs of connection points of the three tested models. From Table 3 we can see that the geometrical distortion of connection points between the original model and the watermarked model is very small. Since only the position of each mark connection point is modified by the watermark embedding, our scheme does not alert the topological relation of the process plant model. Therefore, our scheme is visually and functionally imperceptible.

Table 3 The RMSE values of connection points of each tested model. The number of connection points (CPs) and mark connection points (MCPs) of each model are also listed

Model	CPs	MCPs	RMSE($\times 10^{-4}$)
Carton board	13964	7002	0.237
Hydrogenation	32624	16556	0.512
Styrene	38198	19484	0.403

5.4 Discussion of watermarking targets

Connection points, rather than geometrical parameters of plant components, are selected as watermark carriers in our scheme due to the following reasons.

- First of all, as described in Section 2.1, the topological relation among plant components is represented through connection points. Any malicious attack against topological relation will inevitably give rise to the modification of corresponding connection points.
- Second, geometrical parameters of plant components are employed to support the automatic generation of various construction documents. The modification of geometrical parameters will certainly result in incorrect construction documents. On the contrary, no geometrical and topological information of plant components will be induced by slight coordinates modification of connection points.

Therefore, we conclude that connection points are the best candidates for watermark embedding.

Both theoretical analysis and experimental results discussed above demonstrate that our scheme can resist to various operations provided by Computer-Aided Plant Design systems which may be seen as malicious or non-malicious attacks. However, in theory, one may still change the components while deliberately keep the connection points unchanged through various possible means. For example, an existing component may be deleted or replaced with a new component of the same type while its connection points are deliberately kept unchanged. In that case, the geometrical information of those connection points is kept the same. But the topology constraint is modified since the corresponding plant components they subject to are changed. As discussed above, this kind of attacks can still be detected and located by our scheme.

6 Conclusion and future work

In this paper, we investigate the problem of topology authentication for process plant models. These models, compared with traditional mechanical CAD drawings, feature their intrinsical complex topological relation rather than geometrical shape. We proposed a semi-fragile watermarking scheme to cope with the topology authentication problem. The topological relation among plant components is employed to generate the content-based watermark bits. These topology sensitive watermark bits are then embedded into the similarity transformation invariant of each mark connection point. Both theoretical analysis and experimental results have demonstrated that our scheme has strong ability in detecting and locating various topology attacks. Meanwhile, our scheme is robust against various non-malicious attacks.

There are also some limitations that will motivate our future research. Currently, our scheme can only authenticate the integrity of topological relation of process plant models. However, geometrical parameters of plant components are also crucial for the automatic generation of construction documents, such as isometrics, orthographics, etc. Hence, in our future work, we hope to take both of the geometrical and topological information into consideration for integrity authentication and verification.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China (NO.61170250, NO.61103201). The models used in this paper are the courtesy of Beijing Zhongke Fulong Computer Technology Co., Ltd. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Burdorf A, Kampczyk B, Lederhose M, Schmidt-Traub H (2004) CAPD-computer-aided plant design. Comput Chem Eng 28(1-2):73–81
2. Chang CC, Chen KN, Lee CF, Liu LJ (2011) A secure fragile watermarking scheme based on chaos-and-hamming code. J Syst Software 84(9):1462–1470
3. Coatrieux G, Pan W, Cuppens-Boulahia N, Cuppens F, Roux C (2013) Reversible watermarking based on invariant image classification and dynamic histogram shifting. IEEE T Inf Foren Sec 8(1):111–120
4. Dow MR (1987) Integration of calculation models and CAD systems in building services design. Comput Aided Design 19(5):226–232
5. Feng XQ, Zhang WY, Liu YN (2012) Double watermarks of 3D mesh model based on feature segmentation and redundancy information. Multimed Tools Appl pp 1–19, DOI 10.1007/s11042-012-1039-7
6. Gao XF, Zhang CM, Huang Y, Deng ZG (2012) A robust high-capacity a□ne-transformation-invariant scheme for watermarking 3D geometric models. ACM T Multim Comput 8(S2):34:1–34:21
7. Georgiadisa MC, Macchietto S (1997) Layout of process plants: A novel approach. Comput Chem Eng 21(Supplement 1):S337–S342
8. Guirardello R, Swaney RE (2005) Optimization of process plant layout with pipe routing. Comput Chem Eng 30(1):99–114

9. Hamming RW (1950) Error detecting and error correcting codes. Bell System Technical Journal 26(2):147–160
10. Kwon KR, Chang HJ, Jung GS, Moon KS, Lee SH (2006) 3D CAD drawing watermarking based on three components. In: Proceedings of the IEEE International Conference on Image Processing, Atlanta, GA, USA, pp 1385–1388
11. Kwon KR, Lee SH, Lee EJ, Kwon SG (2006) Watermarking for 3D CAD drawings based on three components. Lect Notes Comput SC 4109:217–225
12. Kwon SH, Kim TW, Choi HI, Moon HP, Park SH, Shin HJ, Sohn JK (2011) Blind digital watermarking of rational Béier and B-spline curves and surfaces with robustness against a□ne transformations and möius reparameterization. Comput Aided Design 43(6):629–638
13. Lang FN, Zhou JL, Cang S, Yu HN, Shang Z (2012) A self-adaptive image normalization and quaternion PCA based color image watermarking algorithm. Expert Syst Appl 39(15):12,046–12,060
14. Lavoué G, Denis F, Dupont F (2007) Subdivision surface watermarking. Comput Graph-UK 31(3):480–492
15. Lee JJ, Cho NI, Lee SU (2004) Watermarking algorithms for 3D NURBS graphic data. EURASIP J Appl Sig P 2004(14):2142–2152
16. Lee SH, Kwon KR (2010) CAD drawing watermarking scheme. Digit Signal Process 20(5):1379–1399
17. Lee SH, Kwon KR (2012) Robust 3D mesh model hashing based on feature object. Digit Signal Process 22(5):744–759
18. Li J, Liu HM, Huang JW, Shi YQ (2012) Reference index-based H.264 video watermarking scheme. ACM T Multim Comput 8(2s):33:1–33:22
19. Li XL, Krishnan S, Ma NW (2010) A wavelet-PCA-based fingerprinting scheme for peer-to-peer video file sharing. IEEE T Inf Foren Sec 5(3):365–373
20. Mooney A, Keating JG, He□ernan DM (2006) A detailed study of the generation of optically detectable watermarks using the logistic map. Chaos Soliton Fract 30(5):1088–1097
21. Ohbuchi R, Masuda H (2000) Managing CAD data as a multimedia data type using digital watermarking. In: Proceedings of the IFIP TC5 WG5.2 Fourth Workshop on Knowledge Intensive CAD to Knowledge Intensive Engineering, Parma, Italy, pp 103–116
22. Ohbuchi R, Masuda H, Aono M (1999) A shape-preserving data embedding algorithm for NURBS curves and surfaces. In: Proceedings of the Computer Graphics International, Alberta, Canada, pp 180–187
23. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572
24. Peng F, Guo RS, Li CT, Long M (2010) A semi-fragile watermarking algorithm for authenticating 2D CAD engineering graphics based on log-polar transformation. Comput Aided Design 42(12):1207–1216
25. Peng F, Lei YZ, Long M, Sun XM (2011) A reversible watermarking scheme for two-dimensional CAD engineering graphics based on improved

difference expansion. Comput Aided Design 43(8):1018–1024
26. Singh J, Garg P, De A (2012) Multiplicative watermarking of audio in DFT magnitude. Multimed Tools Appl 61(2):1–23
27. Wang K, Lavoué G, Denis F, Baskurt A (2008) A comprehensive survey on three-dimensional mesh watermarking. IEEE T Multimedia 10(8):1513–1527
28. Wang K, Lavoué G, Denis F, Baskurt A (2011) Robust and blind mesh watermarking based on volume moments. Comput Graph-UK 35(1):1–19