Robust 2D Engineering CAD Graphics Hashing for Joint Topology and Geometry Authentication via Covariance-Based Descriptors

Zhiyong Su, Ying Ye, Qi Zhang, Weiqing Li, and Yuewei Dai

***Abstract*— This paper investigates the joint authentication of topology and geometry information of 2D engineering computer-aided design graphics, which focus more on topological modeling than geometric modeling of objects. A robust hashing scheme is proposed for joint topology and geometry authentication. The covariance matrices of descriptors are explored to fuse and encode both topology and geometry features of different types into a compact representation. First, a normalized binary shape texture is rendered for each geometric object through the render-to-texture technique. Then, for each geometric object, geometry features are computed based on statistical features that are extracted from image rings. Additionally, topology features are generated according to the topological relations among joint objects. To generate hash codes of the graphic, all geometric objects are first grouped according to their geometry features. Then, for each group, the covariance matrices of descriptors are applied to fuse both the topology and geometry features of all objects, and the intermediate hash codes of each group are computed based on the covariance matrices. The final hash sequence is formed by concatenating the intermediate hash codes that correspond to each group. Secret keys are introduced into both feature extraction and hash construction. The hashes are robust against topology-preserving graphic manipulations and sensitive to malicious attacks. By decomposing the hashes, the locations of tampered objects can be determined. Experimental results are presented to evaluate the performance and show the effectiveness of the method.**

***Index Terms*— Covariance descriptor, authentication, topology authentication, geometry authentication, hash.**

I. Introduction

ENGINEERING computer-aided design (CAD) graphics are a very important type of industrial graphical documentation and are extensively used in Architecture, Engineering and Construction (AEC), which is one branch of CAD. With intensive global competition and increasing product complexity in the AEC industry, companies are increasingly focusing on collaborative design technologies in which a company concentrates only on its core activity and collaborates with other companies for other activities. These technologies provide a consistent set of solutions to support the collaborative creation, management, dissemination, and use of design documentation throughout the entire product and project lifecycle [1]. Therefore, the integrity and security of engineering CAD graphics sharing among all collaborative participants are essential for successful Product Lifecycle Management (PLM) applications.

The digital contents of engineering CAD graphics typically consist of geometry, engineering, and topology information. Geometry information refers to the shape, dimensions and position of objects. Geometric shapes of objects can be designed by using basic geometric entities, such as LINE, POLYLINE, ARC, CIRCLE and 3DFACE. Engineering information refers to design constraints, engineering disciplines, etc. Topology information describes the complex topological relations among various joint objects. The design of engineering CAD graphics focuses on topological modeling more than geometric modeling of objects. The objective of topological modeling is to determine the most economical spatial arrangement of various objects that satisfies construction, operation, maintenance, and safety requirements [2], [3]. This is significantly different from traditional mechanical CAD, as another branch of CAD, which focuses on geometric modeling. Both topology and geometry information should be taken into consideration in content authentication.

Content authentication and identification techniques can be classified into two main categories from the technological perspective: watermarking and hashing. In watermarking-based techniques, watermarks that are associated with authentication information are embedded into specific areas of the content and then extracted to judge whether there have been malicious manipulations of the received content. The precision of the host content is inevitably changed slightly by watermarking [4], [5]. This is an important problem in highly detailed digital design graphics in CAD applications. Different from watermarking-based techniques, hashing-based schemes require no embedding process. Hash codes are generated based on well-designed features that are extracted from the host content and are in accordance with certain characteristics. Content authentication is performed by comparing the hash codes of the host content with the hash codes of the received

Manuscript received June 25, 2017; revised October 6, 2017 and November 15, 2017; accepted November 16, 2017. Date of publication November 23, 2017; date of current version January 3, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61300160. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefano Tubaro. *(Corresponding author: Zhiyong Su.)*

Z. Su, Y. Ye, Q. Zhang, and Y. Dai are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: suzhiyong@njust.edu.cn; 2267008861@qq.com; qiqizhang925@126.com; daiywei@163.com).

W. Li is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: li_weiqing@139.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2777341

content [6]–[8]. Therefore, hashing-based techniques do not introduce any distortion into the host content and are generally more suitable for CAD applications.

To the best of our knowledge, a detailed analysis of the authentication of both topology and geometry information for 2D engineering CAD graphics has not been reported in the literature. In the case of geometry authentication, many digital watermarking schemes have been recently proposed for mechanical CAD graphics [4], [9]–[12], and multiple hashing-based authentication schemes have been proposed for vector data models [13], [14]. By comparison, few related works on topology authentication have been reported. The topology authentication problem of piping isometric drawings, which are a type of 2D engineering CAD graphics, was introduced by Su et al. [15], and a watermarking-based scheme was proposed to verify only the topological integrity. The problem of joint topology and geometry authentication for 2D engineering CAD graphics has yet to be addressed.

A. Contributions

In this paper, we aim to tackle the problem of joint topology and geometry information authentication for 2D engineering CAD graphics. The contributions of this paper can be summarized as follows:

(1) A novel framework for jointly authenticating topology and geometry information of 2D engineering CAD graphics is proposed in this paper. The framework decomposes the authentication task into three stages: topology and geometry feature extraction, topology and geometry feature fusion, and joint topology and geometry hashing.

(2) The geometry features of each geometric object are extracted from a normalized texture through ring partitioning [8], [16]. The normalized texture onto which the geometric object is projected orthogonally is invariant to object translation and uniform scaling. The proposed descriptor is made robust to a wide range of non-malicious manipulations, such as global and local rotation, uniform scaling and translation (RST) transformations, by applying a shape texture rendering method for geometric objects.

(3) Covariance matrices are proposed as a new descriptor for fusion of topology and geometry features. While similar descriptors have been proposed for object tracking and texture analysis in 2D images, this is the first time that covariance-based analysis is explored for content authentication of CAD graphics in the literature. The advantage of using covariance matrices compared with geometric descriptors is that they enable the fusion of multiple and heterogeneous features without the need for normalization [17], [18].

(4) A hashing-based scheme is proposed for authenticating topology and geometry information of 2D engineering CAD graphics. The proposed method is robust to a wide range of non-malicious manipulations, such as global and local RST transformations, while it is also sensitive to topology and geometry changes that are caused by malicious attacks. Furthermore, it can detect and locate tampered objects.

The rest of this paper is organized as follows. Section II reviews the related work. Section III introduces the preliminaries used in this paper. Section IV overviews the framework of the proposed scheme. Details of the proposed hashing scheme are described in Section V, Section VI, and Section VII, respectively. Section VIII presents the performance analysis and experimental results. This work is concluded in Section IX.

II. Related Work

This section reviews some related works on the geometry and topology authentication for CAD models.

A. Geometry Authentication

Existing works on geometry authentication for CAD models in the literature can be divided into two main categories: watermarking-based methods and hashing-based methods.

Watermarking-based methods: Many watermarking-based methods for geometry authentication of CAD models have been reported in the past years [19], [20]. Fornaro et al. [21] proposed a distributed watermarking scheme for verifying Constructive Solid Geometry (CSG) models. Watermarks were computed from selected attributes of the model and stored in control nodes or in the comments of the model. Peng *et al.* [12] presented two reversible watermarking schemes, which can be applied for content authentication, for 2D CAD engineering graphics based on histogram shifting. Both schemes exploited the correlation among adjacent coordinates or relative phases. Watermarks were embedded by shifting and modifying the difference histogram of coordinates or phase. Xiao *et al.* [4] introduced a combined reversible watermarking scheme for 2D CAD engineering graphics. Watermarks were embedded into the distance ratios of vertices through improved quantization index modulation and improved difference expansion.

Hashing-based methods: An information-theoretic hashing of a 3D mesh using spectral graph theory and entropic spanning trees was presented by Tarmissia and Hamza [22]. The scheme applied eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh and then generated the hash value based on the spectral coefficients and the Tsallis entropy estimate. Lee *et al.* [14] proposed a vector data hashing method for authentication and copyright protection of CAD design graphics. Feature values were extracted by projecting the polyline curvatures, which were obtained from groups of vector data using GMM (Gaussian mixture model) clustering, onto random values. The final hash values were generated based on the binarization of feature values.

B. Topology Authentication

The problem of topology authentication for engineering CAD graphics in the AEC industries is relatively new compared with existing image, video, 3D model and vector data hashing and has not been researched as widely compared with geometry authentication. Su *et al.* [15] first investigated the topology integrity authentication problem for piping isometric drawings, which are a type of 2D engineering CAD graphics. A semi-fragile watermarking scheme was proposed to address this interesting issue. The topological relation among joint

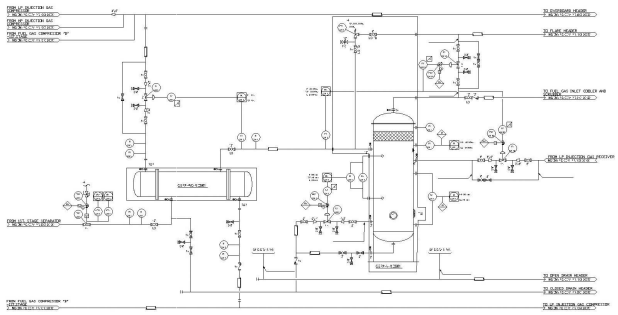

Fig. 1. Part of a typical 2D engineering CAD graphic.

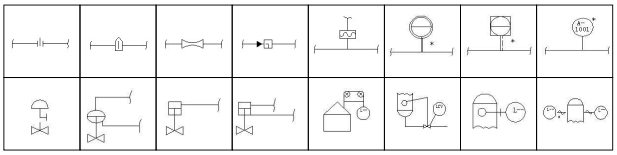

Fig. 2. Some geometric objects used in 2D engineering CAD graphics.

components was encoded into watermarks. Authentication was performed by embedding topology-sensitive watermarks into geometrical invariants of selected objects via quantization index modulation.

Although significant progress has been made in geometry authentication for CAD models, there are still very few methods that focus on topology authentication. Furthermore, the problem of joint topology and geometry authentication for 2D engineering CAD graphics has not been well investigated in the literature. Therefore, this paper aims at developing hashing-based methods for jointly authenticating topology and geometry information for 2D engineering CAD graphics.

III. Preliminaries

A. 2D Engineering CAD Graphics

D engineering CAD graphics consist of multiple geometric objects. Fig. 1 shows part of a common 2D engineering CAD graphic. Fig. 2 shows some typical geometric objects, which are composed of various basic geometric entities such as LINE, POLYLINE, CIRCLE, ARC and POLYGON. These objects often have complex external and internal shapes, as illustrated in Fig. 2. In terms of geometry and topology information, without loss of generality, a 2D engineering CAD graphic $\mathbf{G}$ can be defined as an undirected graph $\mathbb{G} = (\mathbb{O}, \mathbb{E})$, where $\mathbb{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_m\}$ is the set of nodes, and $\mathbb{E} = \{\mathbf{e}_{ij}\}$ is the set of edges. Each node $\mathbf{o}_i$ corresponds to a geometric object. Each edge $\mathbf{e}_{ij} = [\mathbf{o}_i, \mathbf{o}_j]$ indicates that $\mathbf{o}_i$ connects with $\mathbf{o}_j$.

2D engineering CAD graphics can be easily edited through the various geometry and topology operations that are provided by CAD tools. These operations can be classified into non-malicious and malicious operations. Hash codes are expected to be able to survive non-malicious operations and reject malicious tampering to an acceptable extent. Non-malicious operations cover global and local RST transformations. Global RST transformations are performed on the whole graphic to have a better view, while local RST transformations are often applied to individual objects to achieve a satisfactory appearance and fit. These geometry operations are applied to create cleaner and more legible graphics and facilitate the annotation for various objects. They affect the position, dimensions and orientation of objects, based on the precondition of keeping topological relations unchanged. Malicious operations include inserting objects, deleting objects, and changing topological relations logically. The insertion and deletion of objects, which can be defined as malicious geometry attacks, always involve topology modification. All of the above operations are performed on objects, rather than on their geometric entities.

B. Vector Quantization

The Vector Quantization (VQ) technique is utilized to cluster geometric objects in this paper. It was introduced as an image compression technique and proved to be efficient [23].

VQ can be simply regarded as a mapping function that maps the m-dimensional space R^m into a finite subset $Y = \{Y_0, Y_1, \cdots, Y_{k-1}\}$, where Y is called a codebook with k codewords and $Y_j = \{Y_j^0, Y_j^1, \cdots, Y_j^{m-1}\}$ is the j-th codeword in codebook Y. Codebook training is performed in advance through the Linde-Buzo-Gray (LBG) algorithm [24] in this paper. The details of the LBG algorithm are given as follows:

Step 1 : Generate an initial codebook Y^0 of size k. Set the iteration counter $i = 0$ and the initial average distortion $D_{-1} = \infty$. Set the maximum iteration counter as I and the distortion threshold as ε.

Step 2 : For each training vector x, find its best-matching codeword with the least distortion in the current codebook Y^i by calculating the Euclidean distance between each codeword and the input vector x.

Step 3 : Assign the training vectors into k cells and update the centroid of each cell to obtain a new codebook Y^{i+1}.

Step 4 : Calculate the current average distortion D_i for all training vectors at the i-th iteration.

Step 5 : If $(D_{i-1} - D_i)/D_i \leq \varepsilon$ or $i = I$, set the ultimate codebook $Y = Y^{i+1}$ and the LBG algorithm is complete. Otherwise, let $i = i + 1$ and return to Step 2.

C. Covariance Descriptor

The covariance descriptor, which was first introduced by Tuzel *et al.* [17] for object detection and texture classification, is employed to fuse and represent topology and geometry features of 2D engineering CAD graphics in this paper.

From a statistics point of view, covariance can be understood as a measure of how several variables change together. Within the context of the descriptor definition, the set of random variables must correspond to a set of observable features that are correlated to one another [18], [25]. Given an image $I \in R^{W \times H}$, let $F(x, y)$ be the $W \times H \times d$-dimensional feature image that is extracted from I,

$$F(x, y) = \phi(I, x, y) \tag{1}$$

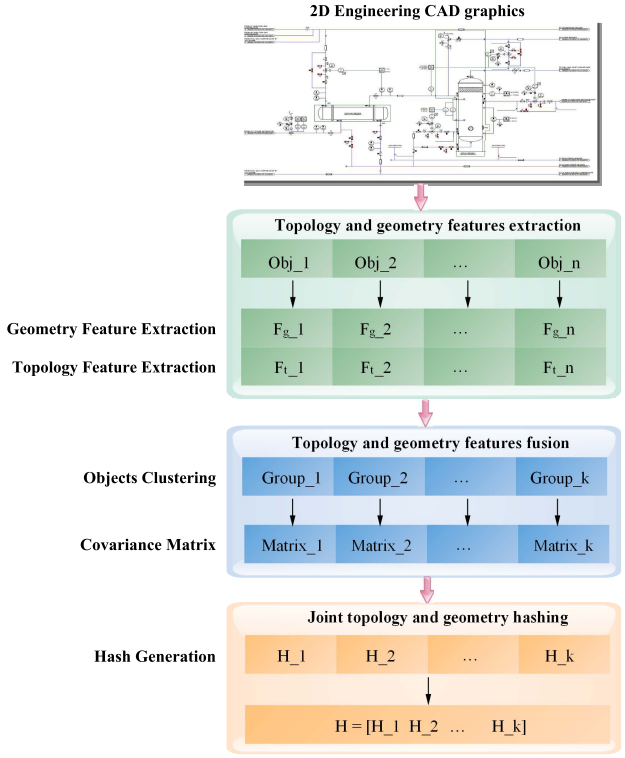

Fig. 3. Overview of the proposed framework.

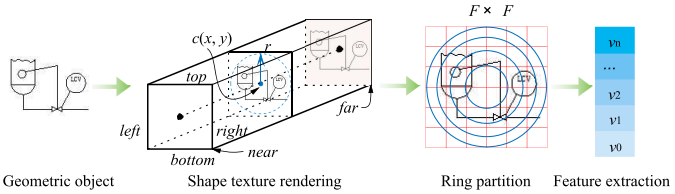

Fig. 4. Illustration of geometry feature extraction.

where the function ϕ can be any pixel-wise mapping, such as intensity, color, gradient, filter response, or higher-order derivative. For a given rectangular region $R \in F$, let $\{z_i\}_{i=1}^{n}$ be the d-dimensional feature points inside R. The region R can be described using a $d \times d$-dimensional covariance matrix of their points [17],

$$C_R = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \mu)(z_i - \mu)^T \tag{2}$$

where μ is the mean of the feature vectors of all points in the region.

IV. Overview of the Framework

The framework of the proposed hashing scheme consists of three major parts: topology and geometry feature extraction, topology and geometry feature fusion, and joint topology and geometry hashing. The flow chart of the authentication framework is shown in Fig. 3.

In the topology and geometry feature extraction part, for each geometric object, a binary shape texture is rendered and its geometry feature is computed based on the ring partition. Its topology feature is extracted according to its topological relation. In the topology and geometry feature fusion part, all objects are clustered into k groups with different numbers of objects according to their geometry features. Then, for each group, a covariance matrix that encodes the topology and geometry features of objects in the group is computed. In the joint topology and geometry hashing part, a feature vector for each group is constructed according to its covariance matrix. To reduce the hash length and for convenience of storage, a Gaussian random matrix is used to compress the feature vector to obtain an intermediate hash, which is then pseudo-randomly scrambled based on secret keys. Encryption and randomization are utilized to reduce hash collisions to improve the security of the algorithm. The final hash sequence is generated by concatenating the intermediate hashes that correspond to the groups.

V. Topology and Geometry Feature Extraction

A. Geometry Feature Extraction

For each geometric object $\mathbf{o}_i$, its geometry feature $\mathbf{v}_i^g$ is computed in the image space, as illustrated in Fig. 4, because of its complex contours and internal structures. A normalized binary texture is first generated by projecting $\mathbf{o}_i$ onto a fixed-size texture orthogonally. Then, the rendered texture is divided into different rings. Finally, its geometry feature $\mathbf{v}_i^g$ is computed based on the statistical features that were extracted from each ring.

1) Shape Texture Rendering: A normalized $F \times F$ binary texture T is rendered for each geometric object $\mathbf{o}_i$ through the Render-To-Texture technique [26], as illustrated in Fig.4. First, an empty texture T is created. Then, the smallest enclosing circle with center $c(x, y)$ and radius r of $\mathbf{o}_i$ is computed. These parameters are further utilized to define the six parameters $(left, right, top, bottom, near, far)$ of the projection matrix, as illustrated in Fig.4. Finally, the object $\mathbf{o}_i$ is rendered to the texture T by orthographic projection. It is obvious that the rendered normalized texture is invariant to object translation and uniform scaling.

2) Ring Partitioning: Ring partitioning [8], [16] is employed to extract geometry features that are robust to object rotation. The rendered normalized texture is divided into a set of rings with equal area, as illustrated in Fig.4. It is theoretically proved that the region in the inscribed circle of an image is preserved under rotation [8], [16]. This provides us with an opportunity to extract image features that are robust to rotation.

Given a normalized $F \times F$ texture T, let n be the number of rings; r_m be the m-th radius ($m = 0, 1, \cdots, n-1$), where the radii are arranged in ascending order; and $\mathbf{R}_m$ be the set of pixel values of the m-th ring. Clearly, $r_{n-1} = \lfloor F/2 \rfloor$ for the texture T. In addition, r_m can be determined by iteratively calculating the following equation:

$$r_m = \sqrt{\frac{\bar{S} + \pi r_{m-1}^2}{\pi}} \tag{3}$$

where

$$r_0 = \sqrt{\frac{\bar{S}}{\pi}} \tag{4}$$

and $\bar{S}$ is the average area of each ring

$$\bar{S} = \lfloor S/n \rfloor \tag{5}$$

in which S is the area of the inscribed circle

$$S = \pi r_{n-1}^2 \tag{6}$$

Thus, image pixels $p(x, y)$ $(0 \le x \le F-1, 0 \le y \le F-1)$ can be classified into different sets by comparing their distances to the image center with these radii

$$\mathbf{R}_0 = \{p(x, y) | d_{x,y} \le r_0\} \tag{7}$$

$$\mathbf{R}_m = \{p(x, y) | r_{m-1} \le d_{x,y} \le r_m\} (m = 1, 2, \cdots, n-1) \tag{8}$$

where $d_{x,y}$ is the Euclidean distance from $p(x, y)$ to the image center (x_c, y_c) which is defined as:

$$d_{x,y} = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{9}$$

where $x_c = y_c = F/2+0.5$ if F is an even number. Otherwise, $x_c = y_c = (F+1)/2$.

3) Feature Extraction: Four statistics are chosen to efficiently capture the visual content of each ring $\mathbf{R}_m$: mean (μ_m), variance (δ_m), skewness (s_m), and kurtosis (w_m), which are defined as follows:

$$\mu_m = \frac{1}{N_m} \sum_{i=0}^{N_m-1} R_m(i) \tag{10}$$

$$\delta_m = \frac{1}{N_m - 1} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^2 \tag{11}$$

$$s_m = \frac{\frac{1}{N_m} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^3}{\left(\sqrt{\frac{1}{N_m} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^2}\right)^3} \tag{12}$$

$$w_m = \frac{\frac{1}{N_m} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^4}{\left(\sqrt{\frac{1}{N_m} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^2}\right)^2} \tag{13}$$

where $N_m = card(\mathbf{R}_m)$ is the total number of elements in $\mathbf{R}_m$, and $R_m(i)$ is the i-th element of $\mathbf{R}_m$ $(0 \le i \le N_m - 1)$. These statistics of each ring are exploited to form the geometry feature vector $\mathbf{v}_i^g$, which contains $(4 \times n)$ elements.

$$\mathbf{v}_i^g = [\mu_0, \delta_0, s_0, w_0, \cdots, \mu_{n-1}, \delta_{n-1}, s_{n-1}, w_{n-1}] \tag{14}$$

B. Topology Feature Extraction

For each object $\mathbf{o}_i$, a fixed-dimensional topology feature vector $\mathbf{v}_i^t$ is formed according to its topological relation

$$\mathbf{v}_i^t = [n_{max}, \mathbf{v}_0^g, \mathbf{v}_1^g, \cdots, \mathbf{v}_{n_{max}-1}^g] \tag{15}$$

where $\mathbf{v}_j^g (0 \le j \le n_{max} - 1)$ is the j-th joint object of $\mathbf{o}_i$ and n_{max} is the maximum number of joint objects. Elements of the geometry feature vectors are set to zero if the object $\mathbf{o}_i$ has less than n_{max} joint objects. The number n_{max} and geometry feature vectors of all joint objects together form a topology feature vector $\mathbf{v}_i^t$, which contains $(1 + n_{max} \times 4 \times n)$ elements.

VI. Topology and Geometry Feature Fusion

A. Objects Clustering

To facilitate tampering localization and ensure that the generated hash has a fixed length and the same computational complexity, for a given 2D engineering CAD graphic $\mathbf{G}$, all objects are clustered into k groups $\{\mathbb{G}_j,\ 0 \le j \le k-1\}$ according to their geometry features using the vector quantization technique [23]. Thus, objects with similar shape are clustered into the same group.

Many geometric objects of 2D engineering graphics are collected and selected to train a codebook Y through LBG [24]. The codebook size is predefined as k, and the influence of the codebook size is analyzed further in Section VIII-C. With VQ, for the geometry feature vector $\mathbf{v}_i^g$ of each object $\mathbf{o}_i$, we find the best-matching codeword Y_j and its index j. Then, we assign object $\mathbf{o}_i$ to the j-th group $\mathbb{G}_j$.

B. Covariance Matrix for Fusing of Geometry and Topology Features

For each group $\mathbb{G}_j$ with n_j objects, a covariance matrix is built for fusing of topology and geometry features of all objects in $\mathbb{G}_j$.

First, a feature selection function $\Phi(\mathbb{G}_j)$ is defined for a given group $\mathbb{G}_j$:

$$\Phi(\mathbb{G}_j) = \{\mathbf{v}_i,\ \forall \mathbf{o}_i \text{ s.t. } \mathbf{o}_i \in \mathbb{G}_j,\ 0 \le i \le n_j - 1]\} \tag{16}$$

where $\mathbf{v}_i$ is the feature vector that encodes the topology and geometry properties of each object $\mathbf{o}_i$, which is defined as:

$$\mathbf{v}_i = [\mathbf{v}_i^g, \mathbf{v}_i^t] = [\mathbf{v}_i^g, n_{max}, \mathbf{v}_0^g, \mathbf{v}_1^g, \cdots, \mathbf{v}_{n_{max}-1}^g] \tag{17}$$

where $\mathbf{v}_i^g$ is the geometry feature vector and $\mathbf{v}_i^t$ is the topology feature vector. The fixed d-dimensional feature vector $\mathbf{v}_i$ is computed for object $\mathbf{o}_i$ of $\mathbb{G}_j$, where $d = 1 + (n_{max}+1) \times 4 \times n$.

Then, a $d \times d$-dimensional Symmetric Positive Definite (SPD) covariance matrix $\mathbf{M}_{\mathbb{G}_j}$ is defined to represent group $\mathbb{G}_j$:

$$\begin{aligned} \mathbf{M}_{\mathbb{G}_j} &= \frac{1}{n_j - 1} \sum_{i=0}^{n_j-1} (\mathbf{v}_i - \mu)(\mathbf{v}_i - \mu)^T \\ &= \begin{bmatrix} m(1,1) & m(1,2) & \dots & m(1,d) \\ \dots & \dots & \dots & \dots \\ m(d,1) & m(d,2) & \dots & m(d,d) \end{bmatrix} \end{aligned} \tag{18}$$

where μ is the mean of the set of feature vectors $\{\mathbf{v}_i\}$ of group $\mathbb{G}_j$. The diagonal elements of the covariance matrix $\mathbf{M}_{\mathbb{G}_j}$ represent the variances of the features, while its non-diagonal elements represent their pairwise correlations. It has a fixed dimension that is independent of the size of the group. Furthermore, the matrix $\mathbf{M}_{\mathbb{G}_j}$ can be computed for any types of features that are encoded in $\mathbf{v}_i$ without normalization or joint probability estimation. Therefore, covariance matrices provide an elegant mechanism for fusing heterogeneous features of arbitrary dimension and scale [18].

VII. Joint Topology and Geometry Hashing

A. Hash Generation

For each group $\mathbb{G}_j$, we zigzag the upper-triangular elements of $\mathbf{M}_{\mathbb{G}_j}$, which is a symmetric positive definite (SPD) matrix, to obtain the following vector:

$$\mathbf{v}_j^m = [m(1,1), \cdots, m(1,d), m(2,2), \cdots, m(2,d), \cdots, m(d,d)] \quad (19)$$

1) Compression and Projection: A Gaussian random matrix $\mathbf{M}^g$ is generated and employed to reduce the dimensionality of the vector $\mathbf{v}_j^m$. To obtain a compressed vector $\mathbf{v}_j^{mc}$, the equation (20) is used to achieve compression and projection:

$$\mathbf{v}_j^{mc} = \mathbf{M}^g \cdot (\mathbf{v}_j^m)^T \quad (20)$$

where $\mathbf{M}^g \in \mathbb{R}^{s \times (d(d+1)/2)}$, $s = \lfloor d(d+1)/2 \times p \rfloor$ in which p is the projection rate, which is selected experimentally. $\mathbf{M}_{u,v}^g$ is a matrix of independent and identically distributed random variables from a Gaussian probability density function with mean 0 and variance $1/s$ [27].

$$\mathbf{M}_{u,v}^g \sim \mathcal{N}(0, \frac{1}{s}) \quad (21)$$

Finally, a compressed s-dimensional vector $\mathbf{v}_j^{mc}$ is generated.

2) Encryption and Randomization: To increase the security of the proposed hashing algorithm, a deterministic chaotic map is employed to generate a chaotic sequence, which is extremely sensitive to initial conditions [7]. The function that is used in this paper is the logistic difference equation:

$$y_{n+1} = a y_n (1 - y_n) \quad (22)$$

where a is the function seed and y_n is the current value of the mapping, which is between 0 and 1, with an initial value y_0. The sequence that is obtained by iterating from the initial value is chaotic when $3.5699456 < a \leq 4$. A mapping value y_0^I will be generated if the logistic function is seeded with a function seed a_0 and an initial value y_0 for I iterations. Let $y = (y_0^I, y_0^{I+1}, \cdots, y_0^{I+s-1})$ be the generated chaotic sequence with the same initial conditions for different iterations. The compressed vector $\mathbf{v}_j^{mc}$ can be randomized by

$$\begin{aligned}\tilde{\mathbf{v}}_j^{mc} &= (\tilde{\mathbf{v}}_{j,0}^{mc}, \tilde{\mathbf{v}}_{j,1}^{mc}, \cdots, \tilde{\mathbf{v}}_{j,s-1}^{mc}) \\ &= (\mathbf{v}_{j,0}^{mc} \times y_0^I, \mathbf{v}_{j,1}^{mc} \times y_0^{I+1}, \cdots, \mathbf{v}_{j,s-1}^{mc} \times y_0^{I+s-1}) \end{aligned} \quad (23)$$

Then, an intermediate binary hash $\mathbf{h}_j$ for each group $\mathbb{G}_j$ is generated through thresholding

$$\mathbf{h}_j = [h(0), \cdots, h(s-1)] \quad (24)$$

where

$$h(i) = \begin{cases} 1, & \tilde{\mathbf{v}}_{j,i}^{mc} > T_j \\ 0, & \tilde{\mathbf{v}}_{j,i}^{mc} \leq T_j, \end{cases} \quad 0 \leq i \leq s-1 \quad (25)$$

$$T_j = \frac{1}{s} \sum_{i=0}^{s-1} \tilde{\mathbf{v}}_{j,i}^{mc} \quad (26)$$

3) Hash Construction: The intermediate hash $\mathbf{h}_j$ of each group $\mathbb{G}_j$ is concatenated to form the final hash sequence, namely $\mathbf{h}$.

$$\mathbf{h} = [\mathbf{h}_1, \cdots, \mathbf{h}_k] \quad (27)$$

It is clear that the length of our hash $\mathbf{h}$ is $(k \times s)$ bits. To guarantee the uniqueness of the final hash and facilitate the authentication stage, the k groups should be arranged in advance. This can be achieved through sorting the codewords in Y according to their vector component values in sequence. By this approach, a sorted codebook Y is achieved. Then, k groups and their hash codes are arranged consequently.

B. Group-Level Tampering Detection and Localization

The proposed hashing scheme is designed to yield group-level tampering detection and localization ability through comparing a distance metric to measure the similarity between the hash values of each group. Regarding malicious geometry and topology modifications, it is sometimes difficult to locate the tampered objects accurately because of the trade-off between compactness of hash codes and sensitivity to malicious tampering. If the hash $\mathbf{h}$ of a trusted engineering CAD graphic $\mathbf{G}$ is available, it is called the reference hash. The hash of a received engineering CAD graphic $\mathbf{G}'$ to be tested, namely $\mathbf{h}'$, is extracted using the above method. An object group can be considered tampered if it contains maliciously modified objects, and the changes in the objects can be measured via distances between hash values of the trusted graphic and the tested graphic in the corresponding group. Here, two graphics with the same contents do not need to have identical geometry information, only topology information, since objects may be modified by topology-preserving operations such as rotation, uniform scaling and translation, as discussed in Section III-A.

The graphic authentication process consists of the following steps:

Step 1: Decompose the received reference hash $\mathbf{h}$ into k groups $\{\mathbf{h}_j\}(j = 0, \ldots, k-1)$ according to the pre-trained codebook Y. Each group has s bits.

Step 2: For the received graphic $\mathbf{G}'$, extract the geometry feature $\mathbf{v}_i^g$ and then the topology feature $\mathbf{v}_i^t$ of each object.

Step 3: Cluster all of the objects into k groups according to their geometry features with the given codebook Y.

Step 4: Compute the covariance matrix $\mathbf{M}_{\mathbb{G}_j'}$ for fusing the topology and geometry features of objects in each group $\mathbb{G}_j'$.

Step 5: Generate the intermediate hash code $\mathbf{h}_j'$ of each group $\mathbb{G}_j'$, and then form the final hash sequence $\mathbf{h}'$.

Step 6: To measure the similarity between group $\mathbb{G}_j$ and group $\mathbb{G}_j'$, the normalized Hamming distance d_{group} is exploited as a metric:

$$d_{group}(j) = \frac{1}{s} \sum_{m=0}^{s-1} |\mathbf{h}_j'(m) - \mathbf{h}_j(m)|^2 \quad (28)$$

where $\mathbf{h}_j'(m)$ and $\mathbf{h}_j(m)$ are the m-th elements of $\mathbf{h}_j'$ and $\mathbf{h}_j$, $0 \leq j \leq k-1$, respectively. Thus, the normalized Hamming distance $D_{graphic}$ for graphic similarity measurement is

TABLE I
NONMALICIOUS OPERATIONS AND PARAMETER VALUES

Operations	Parameters	Number of graphics
Global rotation	90, 180	2
Global scaling	0.5, 2	2
Global translation	100(x), 100(y)	2
Local rotation (20% objects)	90, 180	2
Local scaling (20% objects)	0.5, 2	2
Local translation (20% objects)	1.5(x), -2(y)	2
Total		12

TABLE II
MALICIOUS OPERATIONS AND PARAMETER VALUES

Operations	Parameters	Number of graphics
Inserting objects	5, 25	2
Deleting objects	5, 10	2
Changing topology logically	10	1
Total		5

defined as:

$$D_{graphic} = max(d_{group}(0), d_{group}(1), \cdots, d_{group}(k-1)) \tag{29}$$

Step 7: $\mathbb{G}_j$ and $\mathbb{G}_j^{'}$ are said to be functionally identical if $d_{group}(j) < T$, where T is a threshold. Otherwise, the group $\mathbb{G}_j^{'}$ is a tampered version of $\mathbb{G}_j$ or is different from $\mathbb{G}_j$. Furthermore, $\mathbf{G}$ and $\mathbf{G}^{'}$ should be considered functionally identical if $D_{graphic} < T$. Otherwise, they are different graphics or one is a tampered version of the other.

VIII. PERFORMANCE ANALYSIS AND EXPERIMENTAL RESULTS

In this section, various experiments are carried out to evaluate the performance of the proposed hashing scheme for 2D engineering CAD graphics in terms of robustness, sensitivity, discriminative capability and security.

A. Graphic Data Sets

Taking the process plant in the AEC industry for example, 40 different 2D engineering CAD graphics with various numbers of objects (including 10 graphics with approximately 50 objects, 10 graphics with approximately 100 objects, 10 graphics with approximately 300 objects, and 10 graphics with approximately 500 objects) are tested. To train the codebook Y, an object database of approximately 106 different kinds of objects, which were collected from many graphics of process plants, is also constructed.

Detailed parameter settings of non-malicious and malicious operations are presented in Tables I and II, respectively. Each test graphic has 12 non-maliciously attacked versions and 5 maliciously attacked versions with different tampering ratios. Therefore, $40 \times 12 = 480$ pairs of identical graphics are used for robustness validation, $40 \times 5 = 200$ pairs of similar graphics are used for sensitivity validation, and $40 \times (40-1)/2 = 780$ pairs of different graphics are used for discrimination testing.

B. Performance Criteria

To discuss the performance in detail, the true positive rate (TPR) P_{TPR} and false positive rate (FPR) P_{FPR} are first defined:

$$P_{\text{TPR}} = \frac{N_{\text{similar}}}{N_{\text{identical}}} \tag{30}$$

$$P_{\text{FPR}} = \frac{N_{\text{distinct}}}{N_{\text{different}}} \tag{31}$$

where N_{similar} is the number of pairs of functionally identical graphics that are correctly identified as functionally identical graphics, $N_{\text{identical}}$ is the total number of pairs of functionally identical graphics, N_{distinct} is the number of pairs of distinct graphics that are mistakenly classified as the same graphics, and $N_{\text{different}}$ is the total number of pairs of different graphics. Then, a new term, namely "Detection rate" D_r, is defined to describe the probability of correct detection:

$$D_r = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{32}$$

where N_{correct} is the number of correct detections and N_{total} is the total number of tested graphics.

C. Parameter Setting

To achieve satisfactory performance, the parameters that are used in the proposed hashing scheme are estimated via experiments. In the experiments, the parameters that are used are as follows. The rendered binary shape texture size is 100×100 ($F = 100$). Small F leads to loss of fine details, while large F results in high computational complexity. We choose $F = 100$ as an appropriate trade-off. The number n_{max} in equation (15) is determined in accordance with the specific application area. For example, in our case, n_{max} is set to 4 since the maximum number of joint objects generally will not exceed 4 in the process industry. The logistic function in equation (22) is seeded with the values $a = 4$ and $y_0 = 0.20160614$ for 2000 iterations. The number of groups k is equal to the size of the codebook Y. The normalized Hamming distance in Equation (28) is used to measure the hash distances between corresponding groups. Since the proposed method achieves satisfactory performance for all tested operations when $T \geq 0.2$, we set $T = 0.2$.

1) Group Number & Codebook Size k: To facilitate tamper detection and localization, objects are clustered into k groups according to the codebook Y with k codewords in the preprocessing step for each graphic. The proposed scheme is designed to yield the group-level tamper detection and localization capabilities. Large k will result in fewer objects in each group, thereby giving rise to high tamper detection and localization capabilities. However, total hash length, which depends on k and s, will also increase with the increment of k. Thus, there is a trade-off between total hash length and tamper localization capability. There is also a trade-off among total hash length, sensitivity, and discriminative capability. Generally, compact hash codes include less graphical information, which contributes to stronger robustness. However, the discriminative capability and sensitivity will be weaker. In contrast, hash values of longer length include

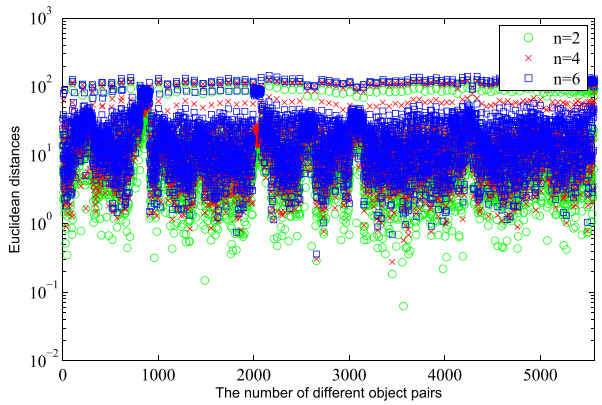

Fig. 5. Euclidean distances between each pair of feature vectors of 106 different kinds of objects.

abundant graphical information. Hence, they contribute to ideal tampering localization functionality. Thus, discriminative capability and sensitivity are stronger and robustness is weaker. In this paper, from the practical application point of view, we set k to 25 as an appropriate trade-off among tamper localization capability, discriminative capability, sensitivity, and total hash length.

2) Ring Number n: Geometry features that are resilient to object translation, scaling, and especially rotation are extracted through dividing the rendered normalized binary texture into n rings. Theoretically, large n will yield better object discrimination performance. However, this will lead to greater geometry feature vector dimension and even higher computational complexity. To select a proper value of n for ring partition, experiments are conducted on the constructed object database with 106 different kinds of objects. The geometry feature vector $\mathbf{v}_i^g$ in equation (14) is first formed for each object for three different values of n ($n = 2, 4, 6$). To measure distances between feature vectors, some typical metrics, such as Euclidean distance and cosine distance, may be employed. In this paper, to reflect the absolute differences among individual numerical features, the Euclidean distance is adopted since cosine similarity is generally used as a metric for measuring the distance when the magnitudes of the vectors do not matter. Thus, for each number, the Euclidean distance between each pair of feature vectors is calculated, and $106 \times (106-1)/2 = 5565$ results are finally obtained. The distribution of these Euclidean distances is illustrated in Fig.5, where the x-axis is the number of different-object pairs and the y-axis is the value of the Euclidean distance. Statistics of Euclidean distances under different ring numbers are also computed and given in Table III. It is observed that the proposed scheme achieves better object discrimination power when $n \geq 4$. Therefore, we choose $n = 4$ as an appropriate trade-off between discrimination performance and computational complexity.

3) Projection Rate p: A $s \times (d(d+1)/2)$ Gaussian random matrix $\mathbf{M}^g$ is employed to reduce the dimensionality of the vector $\mathbf{v}_j^m$, which is derived from the covariance matrix $\mathbf{M}_{\mathbb{G}_j}$ in Eq.(20), where $s = \lfloor d(d+1)/2 \times p \rfloor$ and $d = 1 + (n_{max}+1) \times 4 \times n$. Therefore, the projection rate p determines the dimension of the compressed vector $\mathbf{v}_j^{mc}$ and

TABLE III

STATISTICS OF EUCLIDEAN DISTANCES BASED ON 106 DIFFERENT OBJECTS

Number of rings	Min	Max	Mean	Std Dev
n=2	0.0627	121.5968	12.5148	18.4457
n=4	0.2771	126.2303	17.9501	21.5741
n=6	0.3615	146.1591	22.7178	25.0350

TABLE IV

DIFFERENT PROJECTION RATES AND AUTHENTICATION THRESHOLDS FOR ROC CURVES

Items	Values
Projection rates p	3%, 5%, 8%, 10%, 12%
Authentication thresholds T	$m/30(m = 1, 2, 3, \cdots, 30)$

the hash length. The receiver operating characteristic (ROC) graph in which the x-axis is P_{FPR} and the y-axis is P_{TPR} is employed to make visual classification comparisons with respect to robustness and discrimination under different projection rates with $k = 25$. To comprehensively describe the effect of the projection rate on the hash performances under different thresholds, for each projection rate p, different thresholds T are used to find P_{TPR} and P_{FPR}. The ROC curve is finally formed by a set of points with coordinates (P_{FPR}, P_{TPR}). It is clear that P_{FPR} and P_{TPR} are indicators of robustness and discrimination capability, respectively. For two ROC curves, the curve that is close to the top-left corner has better classification performance than the curve that is far away from the top-left corner.

In the experiment, 40 test engineering CAD graphics, as described in Section VIII-A, are used for testing: $40 \times 12 = 480$ pairs of identical graphics for robustness validation and $40 \times (40-1)/2 = 780$ pairs of different graphics for the discrimination test. Table IV presents the values of the projection rate p and threshold T that are used to calculate the ROC curves. For each pair of graphics, the hash code $\mathbf{h}_j^{'}$ of each group $\mathbb{G}_j^{'}$ of test graphic $\mathbf{G}^{'}$ is first extracted. Then, the group distance $d_{group}(j)$ between $\mathbb{G}_j^{'}$ and $\mathbb{G}_j$ is calculated. Finally, the graphic distance $D_{graphic}$ between $\mathbb{G}^{'}$ and its trusted graphic $\mathbf{G}$ is generated. Fig.6 illustrates the ROC curve comparisons among different projection rates. It is observed that all ROC curves are very close to the top-left corner. This means that the proposed hashing scheme has satisfactory classification performance with respect to robustness and discrimination. Moreover, the ROC curve for $p = 0.08$ is slightly closer to the top-left corner than those for other p values. Therefore, a moderate projection rate, e.g., $p = 0.08$, is a good choice for obtaining a desirable trade-off between robustness and discrimination.

4) Authentication Threshold T: The threshold T is utilized to measure the similarity between group and graphic pairs. The smaller the T value is, the better the discriminative capability. However, robustness performance will be degraded as T decreases. Therefore, the threshold T should be chosen according to the specific application, to obtain a satisfactory balance between discrimination and robustness.

To determine the threshold T for differentiating two groups, $40 \times 12 = 480$ pairs of identical graphics and $40 \times 5 = 200$

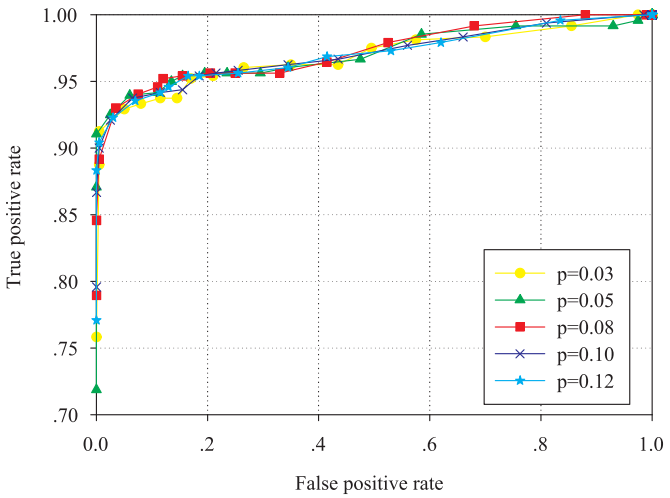

Fig. 6. ROC curve comparisons among different projection rates.

TABLE V
STATISTICS OF NORMALIZED HAMMING DISTANCE BASED ON GRAPHIC PAIRS

Graphic pairs	Min	Max	Mean	Std Dev
Identical graphic pairs	0.000	0.487	0.045	0.092
Similar graphic pairs	0.065	0.547	0.318	0.102
Different graphic pairs	0.472	0.577	0.508	0.020

pairs of similar graphics are used. For each pair of graphics, the hash sequence $\mathbf{h}'$ of each test graphic $\mathbf{G}'$ is first extracted. Then, the graphic distance $D_{graphic}$ between $\mathbb{G}'$ and its trusted graphic $\mathbf{G}$ is calculated. Figs. 7(a) and .7(b) show the normalized Hamming distance distributions for hashes of identical graphics and similar graphics, respectively. Table V illustrates the statistics of normalized Hamming distances. It can be observed that the mean distance of identical graphic pairs is only 0.045 and all maximum distances are less than 0.2, except for those of some rotated graphics. Likewise, the mean distance of similar graphic pairs is 0.318 and all minimum distances are larger than 0.2, except for those of some tampered complex graphics with more than 300 objects. Furthermore, the tampering rates of those graphics range from approximately 1% to 5%. Fig. 8 shows the detection rates of identical and similar graphics under different values of threshold T. When $T = 0.2$, the proposed scheme can achieve a good balance between discrimination and robustness. In this case, 94.58% of identical graphics (including some rotated versions) and 88.50% of similar graphics with low tampering rates can be correctly detected. This is why we set the authentication threshold in subsequent experiments to $T = 0.2$.

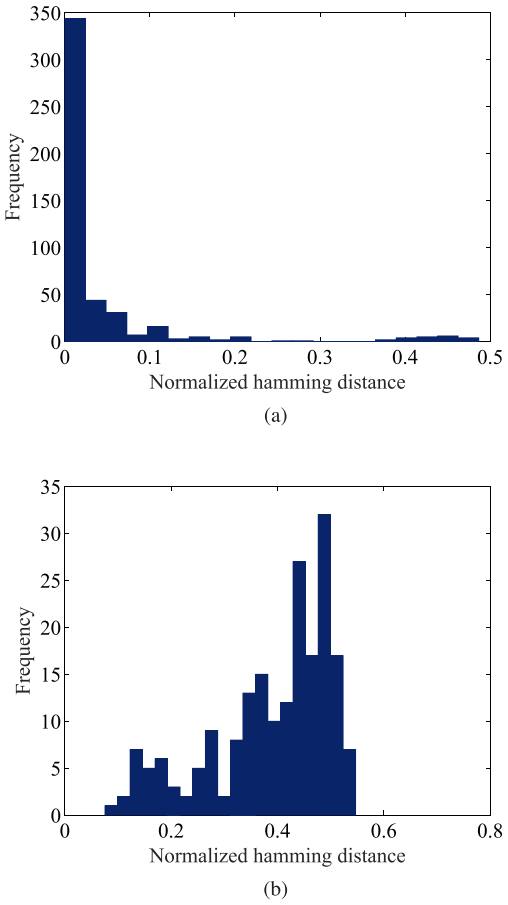

Fig. 7. Distribution of normalized Hamming distances of identical and similar graphic pairs. (a) Identical graphic pairs. (b) Similar graphic pairs.

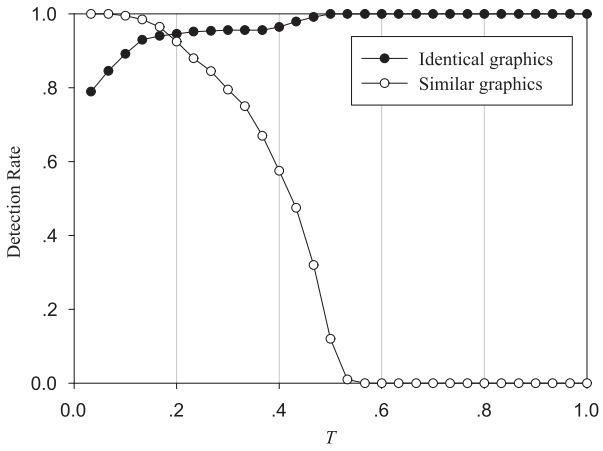

Fig. 8. Detection rates of identical graphics and similar graphics under different thresholds.

D. Robustness Analysis

The proposed hashing scheme is designed to be robust to non-malicious operations, including global and local RST transformations. Under the premise of preserving the topological relationships among objects, these manipulations are performed on graphic objects to obtain a better view or achieve a satisfactory appearance or fit, as discussed in Section III-A. Therefore, these operations only affect the geometric shapes and position of objects.

The test graphics that are used in Section VIII-C.4 are selected for this experiment, and all of the non-malicious operations that are listed in Table I are exploited to attack these graphics. Therefore, each test graphic has 12 functionally consistent graphics and the total number of pairs of identical graphics is $40 \times 12 = 480$. Hash values of the original and

TABLE VI
DETECTION RATES UNDER VARIOUS NONMALICIOUS OPERATIONS LISTED IN TABLE I (%)

Global rotation	Local rotation	Global & Local scaling	Global & Local translation	Total
72.50	95.00	100.00	100.00	94.58

TABLE VII
DETECTION RATE UNDER VARIOUS MALICIOUS OPERATIONS LISTED IN TABLE II (%)

Inserting objects	Deleting objects	Changing topology logically	Total
71.25	100.00	100.00	88.50

the attacked graphics are calculated. Then, the normalized Hamming distance is employed to evaluate their distances. Fig. 7(a) shows the distributions of the calculated normalized Hamming distances. Table VI lists the detection rates under various non-malicious operations. When $T = 0.2$, 94.58% of identical graphics can be correctly detected. Moreover, the mean distance of identical graphic pairs is only 0.045, and all maximum distances are less than 0.2, except those of some rotated graphics. This means that our hashing scheme can achieve satisfactory robustness performance when $T = 0.2$.

E. Sensitivity Analysis

The proposed scheme must be highly sensitive to malicious operations, including inserting objects, deleting objects, and changing topological relations logically. In terms of object addition, the added objects should be connected with existing objects. This kind of attack changes the topology of the modified objects. In the case of object removal, the target objects are first disconnected from their joint objects and then deleted from the graphic. Thus, the topological relation of the involved objects is modified. Modifying local topological relations of objects involves various operations, such as disconnecting two joint objects logically and connecting two disconnected objects logically. Thus, all of the above operations inevitably alter the geometry or topology information of the manipulated objects. Moreover, they lead to the modification of the covariance matrix of the corresponding group and, finally, the generated hash codes.

To further validate the sensitivity of the proposed scheme, the malicious operations that are listed in Table II are used to conduct attacks on each original graphic. Thus, each test graphic has 5 maliciously attacked versions and $40 \times 5 = 200$ pairs of similar graphics are used in total. Finally, 200 normalized Hamming distances are calculated, as shown in Fig. 7(b). Table VII lists the detection rates under various malicious operations. Almost all distances are greater than 0.2, except for those that correspond to some graphics where the tampering ratios are less than 5%. Moreover, only 11.50% of similar graphics with low tampering rates are detected by mistake. This confirms that the proposed method is sensitive to malicious operations.

F. Visual Effect of Tampering Localization

For a content authentication scheme, the tampering localization functionality is of crucial importance. This functionality refers to the capability of identifying tampered graphic objects. The proposed hashing scheme is designed to achieve group-level tampering localization capability, which can be improved by increasing the group number k, as discussed in Section VIII-C.1. A graphic is selected for demonstrating the functionality via the visual effect. Due to space limitations, Fig .9(a) shows only part of the test graphic. The three malicious operations that are discussed above are used to alter graphic objects. The proposed hashing scheme is applied to the test graphic, and it is observed that all normalized Hamming distances are greater than 0.2. All of the object groups to which the tampered objects belong are correctly identified. Then, objects in the identified groups are marked as suspicious objects. Figs. 9(b), (c) and (d) show the visual detection results of the proposed method for different attack types. The detected results are highlighted by red-colored squares. For example, in Fig. 9(b), a new object A of the same type as B_1 is added and connected with B_1 and C_1. Thus, the geometry information of the graphical and topological relations of B_1 and C_1 is modified. The groups to which the attacked objects belong are correctly detected by the proposed scheme. In addition, all of the objects (including A, B_1, B_2, B_3, C_1, and C_2) in the detected groups are labeled and highlighted by red-colored squares.

G. Discriminative Capability Analysis

Discriminative capability means that two different graphics have a very low probability of generating similar hashes. If the normalized Hamming distance between two different graphics is less than the threshold T, a collision occurs.

To evaluate the discriminative capability of the proposed scheme, $40 \times (40-1)/2 = 780$ pairs of different graphics are employed. The proposed hashing scheme is used to extract hashes of 40 different graphics. Then, the normalized Hamming distance $D_{graphic}$ between each pair of different graphics is calculated, and $40 \times (40-1)/2 = 780$ results are finally obtained. The statistics of the normalized Hamming distances of different graphic pairs are listed in Table V. Fig. 10 gives the normalized Hamming distance distributions for hashes of different graphics. According to the results, the minimum and mean distances are 0.472 and 0.508, respectively. Clearly, all distances are much larger than the above mentioned threshold $T = 0.2$, which indicates that the proposed hashing scheme achieves good discrimination.

H. Security Analysis

Security depends on the unpredictability of hash codes. This implies that it should be very difficult to decode a hash without knowledge of the key. The security of the proposed hashing scheme can be guaranteed by applying key-dependent encryption in the process of feature vector compression and randomization. The Gaussian random matrix $\mathbf{M}^g$, which is employed to reduce the dimensionality of the feature vectors, can be kept as a security key. The function seed a and the initial value y_0 in the logistic mapping equation (22), which is utilized to encrypt the compressed feature vectors, can also serve as security keys.

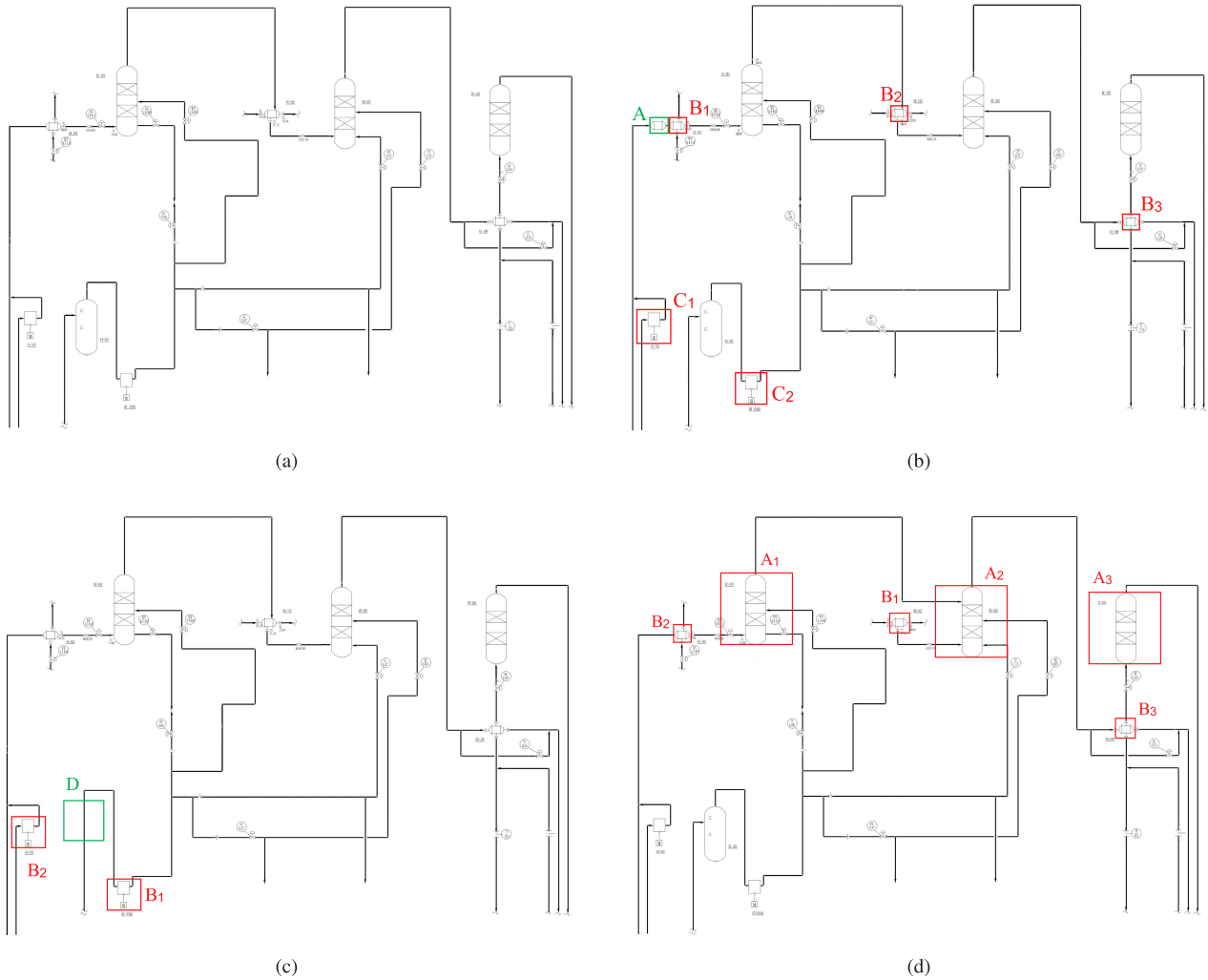

Fig. 9. Tampered graphics and localized objects. (a) Part of the test graphic. (b) An object A is added and connected with B_1 and C_1. (c) An object D is deleted from the graphic. (d) Topology relation among A_1, B_1, and A_2 is modified logically.

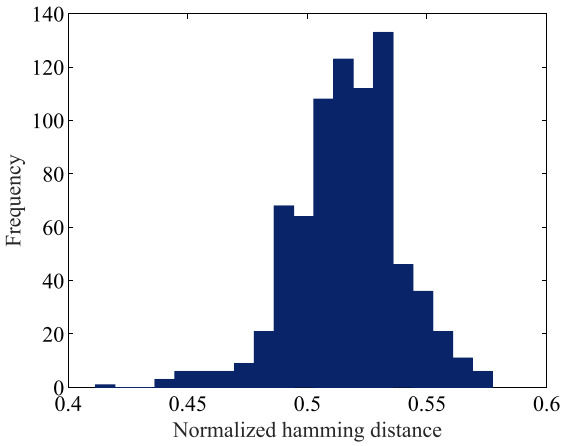

Fig. 10. Distribution of normalized Hamming distances of different graphic pairs.

To validate the security performance of the proposed hashing scheme, in our experiments, 40 test graphics are adopted. Different keys are exploited to extract hashes, and distances between these key-based hashes are calculated. Only secret keys are varied and other parameters remain unchanged. For each graphic, first, a Gaussian random matrix $\mathbf{M}^g$, a function seed a and an initial value y_0 are used to extract the graphic hash. Then, 99 sets of different keys, including Gaussian random matrices, function seeds, and initial values, are employed to generate 99 different graphic hashes. Finally, the normalized Hamming distances between the first hash and the other 99 hashes are computed. Therefore, $40 \times 99 = 3960$ normalized Hamming distances are obtained for the 40 test graphics in total. Fig. 11 shows the obtained results for all test graphics, where the x-axis is the index of the normalized Hamming distance between two hash codes and the y-axis is the normalized Hamming distance. It is observed that the minimum distance is much larger than $T = 0.2$. These results empirically verify that our graphic hashing scheme is key-dependent and meets the security requirements.

I. Fusion Method Analysis

The covariance descriptor is employed to fuse topology and geometry features of each group $\mathbb{G}_j$ in this paper. Compared with other fusion methods, such as feature concatenation, its advantages can be summarized as follows: First, it provides an elegant mechanism for fusing heterogeneous features

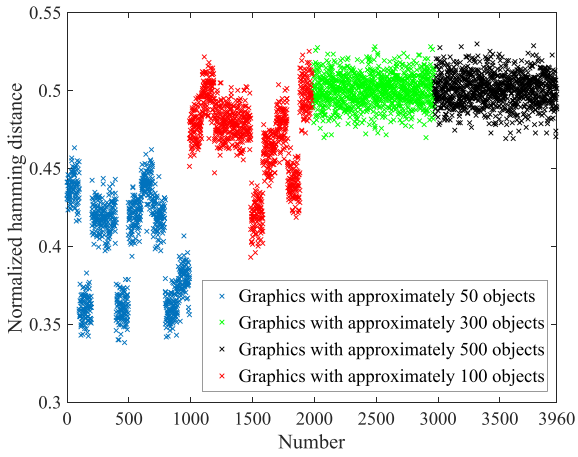

Fig. 11. Distribution of normalized Hamming distances with different keys.

of arbitrary dimension and scale. It captures not only the geometry but also the topology features of objects in each group, thereby characterizing the graphic. Second, it has a fixed dimension that is independent of the size of group $\mathbb{G}_j$. Third, it is compact and easy to compute. Owing to the symmetry, a covariance matrix has a smaller number of distinct elements compared with many other region descriptors.

J. Comparison With Previous Works

To the best of our knowledge, no related work that focuses on authenticating both geometry and topology information of 2D engineering CAD graphics has been reported in the literature. As described in Section II, existing works typically concentrate on geometry authentication and protection for traditional mechanical CAD graphics, and there are still very few methods that focus on topology authentication for 2D engineering CAD graphics [15], [28]. The main advantage of the proposed scheme, compared with previous works on 2D engineering CAD graphics [15], [28], can be summarized as follows:

First, the proposed method can authenticate both geometry and topology information for 2D engineering CAD graphics. Second, the proposed hashing-based method does not introduce any distortion into the original graphics; thus, it is more suitable for CAD applications. The watermarking-based schemes that were presented in [15], [28] were designed to authenticate only the topology information for 2D engineering CAD graphics, and original graphics were inevitably changed by watermarking. Therefore, it is believed that the proposed method is more generic and practical for industrial applications.

IX. Conclusions

In this paper, a novel robust hashing scheme is proposed for jointly authenticating topology and geometry information of 2D engineering CAD graphics. A new covariance-based descriptor is introduced for fusing multiple heterogeneous topology and geometry features. Hashes that are produced with the proposed method are robust to non-malicious operations and are sensitive to changes that are caused by malicious attacks. The hashing scheme that is described in this paper yields group-level tampering detection and localization ability. The hash can be used to differentiate similar and different graphics. It can also identify and locate object groups that contain maliciously attacked objects. The proposed scheme achieves a trade-off among robustness, sensitivity, discriminative capability, and tampering localization. The experimental results show the effectiveness and availability of the proposed hashing algorithm.

Further study is needed to find geometry features that better represent various kinds of geometric objects, to enhance the hash's robustness against the rotation operation. Another important aim for future research is the achievement of more precise tampering localization accuracy while maintaining a short hash length and good sensitivity to malicious attacks.

Acknowledgment

The authors would like to acknowledge the helpful comments and kindly suggestions provided by anonymous referees.

References

[1] W. Shen *et al.*, "Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review," *Adv. Eng. Inform.*, vol. 24, no. 2, pp. 196–207, 2010.
[2] A. Burdorf, B. Kampczyk, M. Lederhose, and H. Schmidt-Traub, "CAPD—Computer-aided plant design," *Comput. Chem. Eng.*, vol. 28, nos. 1–2, pp. 73–81, 2004.
[3] R. Guirardello and R. E. Swaney, "Optimization of process plant layout with pipe routing," *Comput. Chem. Eng.*, vol. 30, no. 1, pp. 99–114, 2005.
[4] D. Xiao, S. Hu, and H. Zheng, "A high capacity combined reversible watermarking scheme for 2-D CAD engineering graphics," *Multimedia Tools Appl.*, vol. 74, no. 6, pp. 2109–2126, 2015.
[5] C.-P. Yan, C.-M. Pun, and X.-C. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Process.*, vol. 121, pp. 1–16, Apr. 2016.
[6] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 55–63, Jan. 2013.
[7] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1336–1349, Jul. 2015.
[8] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 200–214, Jan. 2016.
[9] R. Ohbuchi and H. Masuda, "Managing CAD data as a multimedia data type using digital watermarking," in *Proc. 4th Workshop Knowl. Intensive CAD Knowl. Intensive Eng. (IFIP)*, Deventer, The Netherlands, 2001, pp. 103–116.
[10] S.-H. Lee and K.-R. Kwon, "CAD drawing watermarking scheme," *Digit. Signal Process.*, vol. 20, no. 5, pp. 1379–1399, 2010.
[11] L. Cao, C. Men, and R. Ji, "Nonlinear scrambling-based reversible watermarking for 2D-vector maps," *Vis. Comput.*, vol. 29, no. 3, pp. 231–237, 2013.
[12] F. Peng, Y. Liu, and M. Long, "Reversible watermarking for 2D CAD engineering graphics based on improved histogram shifting," *Comput.-Aided Des.*, vol. 49, no. 4, pp. 42–50, 2014.
[13] S.-H. Lee, S.-G. Kwon, and K.-R. Kwon, "Robust hashing of vector data using generalized curvatures of polyline," *IEICE Trans. Inf. Syst.*, vol. E96.D, no. 5, pp. 1105–1114, 2013.
[14] S.-H. Lee, W.-J. Hwang, and K.-R. Kwon, "Polyline curvatures based robust vector data hashing," *Multimedia Tools Appl.*, vol. 73, no. 3, pp. 1913–1942, 2014.
[15] Z. Su, X. Yang, G. Liu, W. Li, and W. Tang, "Topology authentication for piping isometric drawing," *Comput.-Aided Des.*, vol. 66, no. 9, pp. 33–44, 2015.
[16] Z. Tang, X. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Process.*, vol. 93, no. 7, pp. 2061–2069, 2013.

[17] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. 9th Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 589–600.
[18] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1591–1603, Sep. 2015.
[19] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1513–1527, Dec. 2008.
[20] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Inf. Sci.*, vol. 279, no. 20, pp. 251–272, 2014.
[21] C. Fornaro and A. Sanna, "Public key watermarking for authentication of CSG models," *Comput.-Aided Des.*, vol. 32, no. 12, pp. 727–735, 2000.
[22] K. Tarmissi and A. B. Hamza, "Information-theoretic hashing of 3D objects using spectral graph theory," *Exp. Syst. Appl.*, vol. 36, no. 5, pp. 9409–9414, 2009.
[23] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2, pp. 4–29, Apr. 1984.
[24] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, Jan. 1980.
[25] P. Cirujeda, Y. D. Cid, X. Mateo, and X. Binefa, "A 3D scene registration method via covariance descriptors and an evolutionary stable strategy game theory solver," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 306–329, 2015.
[26] D. Shreiner, G. Sellers, J. Kessenich, and B. Licea-Kane, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, D. Shreiner, Ed., 8th ed. Reading, MA, USA: Addison-Wesley, 2013.
[27] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
[28] Z. Su, L. Zhou, Y. Mao, Y. Dai, and W. Tang, "A unified framework for authenticating topology integrity of 2d heterogeneous engineering cad drawings," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 20663–20689, 2017.

Zhiyong Su received the B.S. and M.S. degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology, in 2004 and 2006, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2009. He is currently an Associate Professor with the School of Automation, Nanjing University of Science and Technology, China. His current interests include computer graphics, computer vision, and argument reality.

Ying Ye received the B.S. degree from Yangzhou University, China, in 2015. She is currently pursuing the M.S. degree with the Nanjing University of Science and Technology, China. Her research interests include image processing and computer vision.

Qi Zhang received the B.S. degree from Anhui University, China, in 2015. She is currently pursuing the M.S. degree with the Nanjing University of Science and Technology, China. Her research interests include image and video processing and computer vision.

Weiqing Li received the B.S. and Ph.D. degrees from the School of Computer Sciences and Engineering, Nanjing University of Science and Technology, in 1997 and 2007, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His current interests include computer graphics and virtual reality.

Yuewei Dai received the M.S. and Ph.D. degrees from the Nanjing University of Science and Technology, in 1987 and 2002, respectively, all in automation. He is currently a Professor with the School of Automation, Nanjing University of Science and Technology, China. His research interests are in the areas of information security, signal, and image processing.