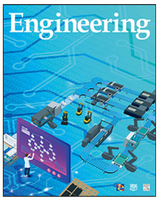
Engineering 7 (2021) 787–797

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng



Research
Intelligent Manufacturing—Article

A Novel Attribute-Based Encryption Approach with Integrity Verification for CAD Assembly Models

Yueting Yang [a], Fazhi He [a,b,*], Soonhung Han [c], Yaqian Liang [a], Yuan Cheng [d]

[a] School of Computer Science, Wuhan University, Wuhan 430072, China
[b] State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[c] Division of Ocean Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[d] School of Information Management, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Article history:
Received 28 November 2019
Revised 16 July 2020
Accepted 19 March 2021
Available online 29 April 2021

Keywords:
Information security
Cloud-based design and manufacture
Co-design
CAD assembly models
Attribute-based encryption

ABSTRACT

Cloud manufacturing is one of the three key technologies that enable intelligent manufacturing. This paper presents a novel attribute-based encryption (ABE) approach for computer-aided design (CAD) assembly models to effectively support hierarchical access control, integrity verification, and deformation protection for co-design scenarios in cloud manufacturing. An assembly hierarchy access tree (AHAT) is designed as the hierarchical access structure. Attribute-related ciphertext elements, which are contained in an assembly ciphertext (ACT) file, are adapted for content keys decryption instead of CAD component files. We modify the original Merkle tree (MT) and reconstruct an assembly MT. The proposed ABE framework has the ability to combine the deformation protection method with a content privacy of CAD models. The proposed encryption scheme is demonstrated to be secure under the standard assumption. Experimental simulation on typical CAD assembly models demonstrates that the proposed approach is feasible in applications.

1. Introduction

As investigated in a survey paper published in *Engineering* [1], intelligent manufacturing is the state-of-the-art product development philosophy, and comprises intelligent manufacturing technology, Internet of Things (IoT)-enabled manufacturing, and cloud manufacturing [1,2]. With the development and application of cloud computing [3–5], traditional computer-aided design (CAD), computer-aided engineering (CAE), and computer-aided manufacturing (CAM) systems are moving to cloud-based design and manufacturing (CBDM) [6–8], raising new information security problems for enterprises, especially for small and medium-sized enterprises (SMEs) [9].

Due to limitations of funds and resources, SMEs cannot build private clouds as large companies do. Well-known public cloud service providers (CSPs) have experienced various security problems; thus, semi-trusted or untrustworthy public clouds challenge the information security of SMEs in cloud-based co-design processes [10–13]. As a core ingredient of collaborative product development [14–18], CAD models contain abundant intellectual properties and therefore encounter security problems. Information security depends on the reliability of the industrial cloud [19,20]. A vital security issue for CBDM is how to avoid illegal access to the confidential information contained in CAD models within co-design environments [21–23].

One typical access-control method is to apply a standard encryption technique, such as attribute-based encryption (ABE) [24]. But a simple introduction of ABE access-control into CAD assembly models is inefficient and inflexible. Some flexible access-control approaches have been reported for cloud computing [25–28]. Therefore, how to develop an efficient and flexible access-control approach to protect outsourced and co-designed CAD assembly models remains a challenge in CBDM.

This paper proposes a novel ABE approach with integrity verification for CAD assembly models. In contrast to existing ciphertext policy ABE (CP-ABE) and related methods [24–28], a hierarchical access-control encryption scheme that enables flexible authentication for users with different privilege levels is proposed for legitimate access to CAD assembly models.

The rest of this paper is organized as follows. In Section 2, related work is reviewed. In Section 3, we discuss our design goals

[*] Corresponding author.
E-mail address: fzhe@whu.edu.cn (F. He).

https://doi.org/10.1016/j.eng.2021.03.011
2095-8099/© 2021 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company.

and declare the problem formulation. In Section 4, we present a cloud-based sharing architecture for assembly models. In Section 5, we elaborate details of the encryption scheme for the proposed approach and illustrate its security proof and theoretical analysis. In Section 6, we demonstrate the performance of the proposed encryption scheme. Finally, the paper is concluded in Section 7.

2. Related work

2.1. Access control

Access control, which originated from the concept of the access matrix, is an important way of protecting data confidentiality and privacy [29–33]. A framework for access control in CAD environments (FACADE) has been presented to protect CAD models [34], and a data-security model for collaborative design and data management systems has been proposed to combine multiple security technologies with access control [35]. Chang et al. [36] reported an access-control system based on multiple methods for use in sharing CAD design drawings. Speier et al. [37] used hybrid access control for product data-security processing.

Roles-based access control is a mainstream approach for co-design [38,39]. However, with the expansion of complex processes, co-design users, and product models, "role explosion" problems can occur. Furthermore, existing access-control methods are based on a single model file and are thus inflexible.

2.2. Encryption approaches

Encryption has been extensively applied in multimedia data [40]. Nishchal and Naughton [41] proposed a multi-level encryption architecture based on optical principles, which can be used to handle three-dimensional (3D) holograms with parallax and multitude sharing. Huang et al. [42] reported an encryption method based on virtual holography for 3D cube data. This method was used to generate holographic images of a 3D cube for computer-simulated holography, based on which the proposed encryption process was carried out. Kim and Yoo [43] and Chen and Tsai [44] proposed a hierarchical encryption method for assembly entity models, in which different users could gain access to data from different parts of an assembly model file. Each level of the file was encrypted with different keys, and each key was only authorized for the use of the designated user. Researchers have also extended the concept of encryption to shape deformation [45–48].

2.3. Attribute-based encryption

Fuzzy identity-based encryption (i.e., ABE) is the most promising encryption primitive for supporting fine-grained access control [24]. At present, ABE is mainly divided into two categories: CP-ABE [28] and key policy ABE (KP-ABE) [49]. In CP-ABE, ciphertext is associated with an access structure defined by the data owner, while keys are associated with attributes. On the contrary, in KP-ABE, ciphertext is associated with attributes while keys are associated with an access structure defined by the data owner. Both schemes are constructed in a single-file scenario.

Hierarchical attribute-based solutions have been studied for multi-file scenarios. Miao et al. [25] introduced the idea of hierarchical data into attribute-based key searching in cloud computing. Wan et al. [26] proposed a hierarchical attribute-set-based encryption (HASBE) scheme built upon an attribute-set-based scheme in order to realize a more precise attributes-satisfaction policy in a multi-file scenario. However, the hierarchical access structure was too complex to apply the scheme for huge CAD models. Wang et al. [27] presented a file hierarchical CP-ABE (FH-CP-ABE), in which a loophole in the recursive formula for ciphertext components might lead to illegal access.

2.4. Contributions

For access-control authentication, this manuscript presents a novel ABE scheme for CAD assembly models, named the CAD assembly hierarchy file CP-ABE (CAD-CP-ABE) scheme. In the proposed technical method, a symmetrical encryption algorithm is utilized for plaintext encryption, and the CAD-CP-ABE scheme is used for content keys (CKs) management.

Unlike existing CP-ABE techniques, the access structure of the upper nodes in the proposed scheme is more concise than that of the lower nodes. Thus, we redefine the node definition and submit a new set of generation rules in order to avoid excessive redundant nodes. In this way, the proposed method makes the hierarchical access structure assembly hierarchy access tree (AHAT) applicable to CAD assembly models. The AHAT contains file nodes, attribute nodes, and threshold nodes. The assembly ciphertext (ACT) file contains the AHAT and all ciphertext elements, and will be used by co-design users with legitimate access rights.

For integrity verification, we propose an assembly Merkle tree (MT) technique after modification of the original MT, which can prevent CAD files from being illegally removed from or added to the cloud service. We also adopt a deformation-based technique to protect shape information. These two methods are combined together to enhance our approach.

3. Problem formulation

Before presenting our architecture (Section 4), we analyze information security problems regarding design chain management (DCM) and collaborative design [38,50–52]. A typical DCM collaboration model is discussed in Ref. [50]. The model establishes six levels for collaboration. Collaboration problems include trusting communication, negotiation, and authority distribution.

In a cloud-based collaboration process, collaborators not only have to negotiate with others, but also have to deal with security problems. Therefore, authentication, integrity, and information privacy are key aspects of information sharing.

Based on the above analysis, we formally define our problems and then indicate our design goals.

3.1. Threat model

The proposed architecture involves four entities: the data owner, data users, authority, and CSP. The data owner is trusted, and the data users are authorized by the authority and are collaborated. The authority is a completely trusted entity that is in charge of generating and distributing public and secret keys (SKs). The CSP is a semi-trusted outsourcing entity in cloud systems; therefore, ① the data could be illegally accessed, and/or ② the cloud services may deviate from the prescribed protocols and mount a data integrity attack. To enforce the confidentiality of information and hierarchy access control, the data owner first deforms the private features of the CAD models and then encrypts the data files with an attribute-based access policy before outsourcing the data files into the CSP. To decrypt the data files shared from various data owners, a data user submits her attributes to the authority in order to obtain her SK.

3.2. Design goals

The proposed approach aims to achieve the following functions and security goals:

- **Authentication.** Each CAD model data file should be accessed through valid access privileges. Our approach ensures that plaintext data cannot be revealed. An adversary cannot learn any useful information from the ACT file. In addition to achieving authorization, we ensure that upper-level privileged users have access to all their lower level privileged user files.
- **Integrity.** A CAD assembly model is usually composed of a group of data files. It is necessary to ensure that none of the files of the model are tampered with, added, or deleted. With ABE, the plaintext of files can be completely hidden. This integrity ensures that the fixed-tree structure of the CAD model is not tampered with by the CSP or by malicious visitors through assembly MT technology.
- **Confidentiality.** Since confidential information is shared with the collaborators, who may be potential competitors, information leakage may occur. This problem is solved by deforming the sketches of CAD models before sharing them.

3.3. Preliminaries

Bilinear maps: Let p be a prime number, G_0 and G_T be two multiplicative groups of integers modulo p. The generator of G_0 is g.

Bilinear mapping e: $G_0 \times G_0 \rightarrow G_T$ satisfies the following properties:

- Bilinearity: For any $u, v \in G_0$ and $a, b \in Z_p$, Z_p = {0, 1, 2, ..., p–1}, it has $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: There exist $u, v \in G_0$, such that $e(u, v) \neq 1$.
- Computability: For all $u, v \in G_0$, there is an efficient computation $e(u, v)$.

Definition 1: Bilinear Diffie-Hellman (BDH) generator. For a random algorithm Γ with a security parameter k ($k > 0$) as input, the BDH generator outputs the description of two groups, G_0 and G_T, and common prime order p. Furthermore, if a bilinear mapping e, $G_0 \times G_0 \rightarrow G_T$, can be effectively calculated in the polynomial time of k, the algorithm is called as a BDH parameter generator.

Definition 2: Determinant BDH (DBDH) problem. Let G_0, G_T, and e be the outputs of the parameter generator Γ in Definition 1, and then let g be the generator of group G_0. The DBDH problem is defined as follows: Given $< g, g^a, g^b, g^c, T >$ (in which the random elements $a, b, c \in Z_p, T \in G_T$), determine whether the equation $e(g, g)^{abc} = T$ established is valid.

MT: In computer science, a hash tree is a tree-like data structure, also known as an MT [53]. As shown in Fig. 1, each leaf node uses the hash of itself as its label (e.g., H3 = hash(A)), while the non-leaf node uses the encrypted hashes of its sub-node labels as its label. An MT can validate the content of a large data structure with the following characteristics:

- The MT is a tree structure that possesses all the tree structure features.
- Without checking all the entire dataset, it can be concisely proved whether a datum belongs to a data group or not.
- The hash of each node in the MT will prove the integrity and correctness of the data content.

4. Overview of the proposed approach

4.1. System architecture

Fig. 2 illustrates the system architecture of our approach for CBDM.

Authority: The authority is a completely trusted entity that can verify the identity attributes of the users.

Data owner: The data owner will store and share a CAD assembly model through CSP. This entity is in charge of creating an assembly MT and an AHAT, deforming sketches, and executing encryption functions. It uploads one model's encrypted files, an ACT file, and an assembly MT to the CSP.

User: The user will download the ACT file and all or parts of encrypted files from one cloud server (B). Cloud server B executes decryption functions. The user will verify the structural integrity of the downloaded files with another cloud server (A).

CSP: The CSP is a semi-trusted outsourcing entity that provides ciphertext storage and transmission services.

4.2. The building blocks of the proposed approach

A secure co-design for CAD assembly models involves three building blocks: the assembly MT, deform-based protection, and the CAD-CP-ABE scheme.

As shown in Figs. 3(a) and (b), an assembly, Assem02, has a fixed structure that can be presented abstractly as a tree graph. Each component (a collective name for the parts and assemblies) becomes a node of this tree graph.

We build an assembly MT to support integrity verification, as shown in Fig. 3. There are two kinds of nodes in the assembly MT. Part (leaf) nodes are represented as [identity (Id), value] pairs, where the value is the hash of the leaf file calculated by a collision-resistant hash function (e.g., message digest algorithm 5 (MD5) and secure hash algorithm (SHA)). Assembly (non-leaf) nodes are represented as [Id, value1, value2] pairs, where value1 is equal to

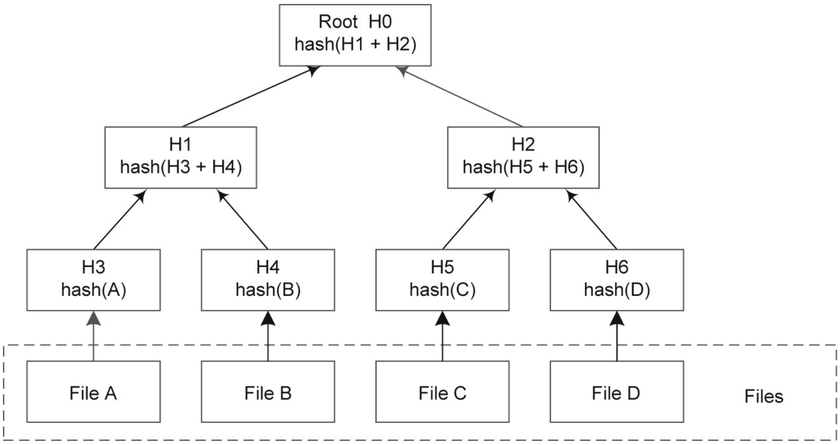


Fig. 1. An example of an MT. A, B, C, and D are four files needed to be encrypted by MT; H0, H1, H2, H3, H4, H5, and H6 are labels of nodes in MT.

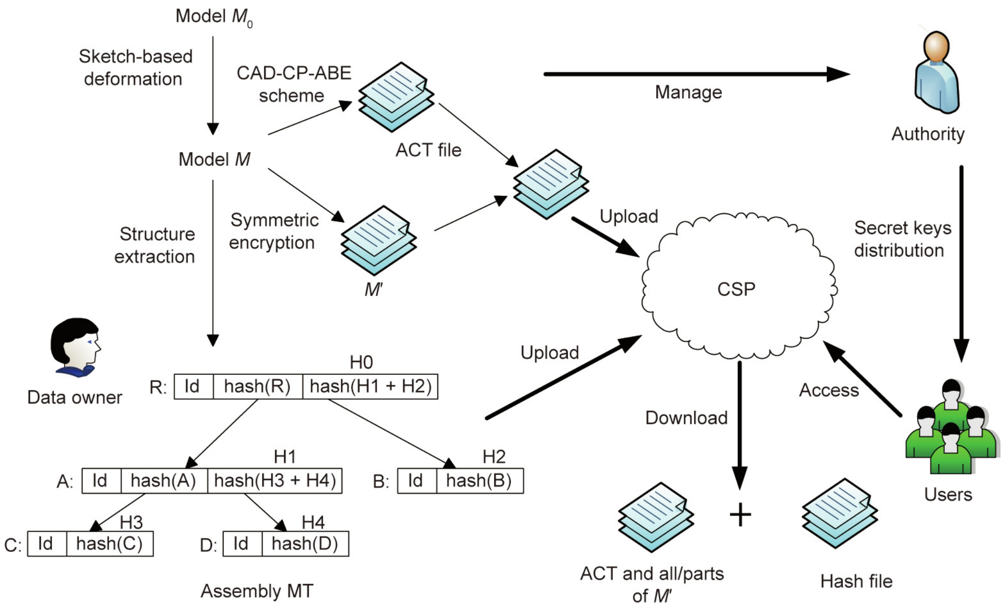


Fig. 2. The architecture of the proposed approach for a CBDM assembly MT. A, B, C, D, and R are five files of M. M_0: original CAD assembly model; M: deformed CAD assembly model; M': ciphertext for M; Id: identity of each file node in assembly MT.

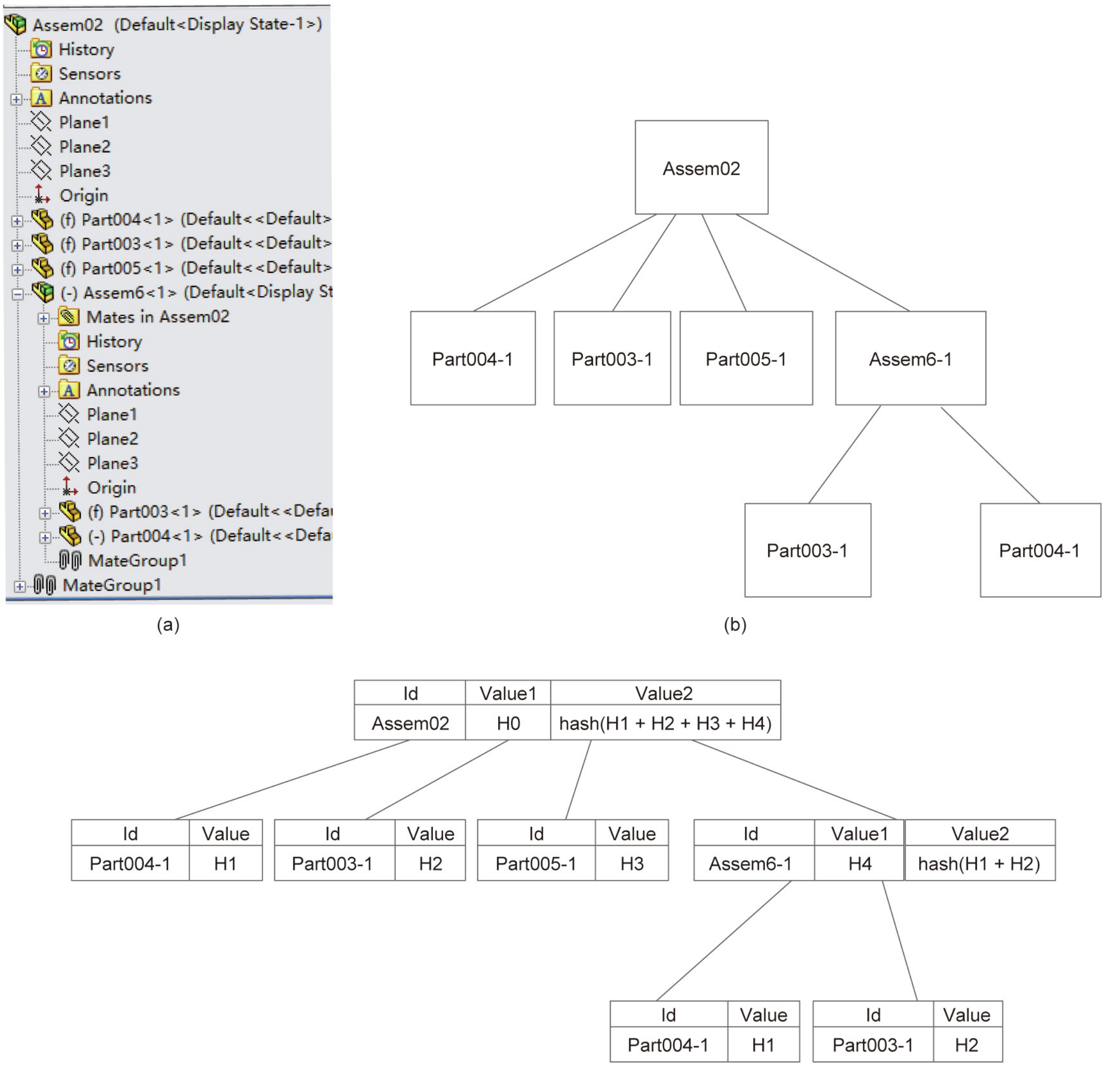


Fig. 3. An illustrative example of Assem02. (a) An assembly structure of Assem02 in SolidWorks; (b) an abstract structure tree of Assem02; (c) the assembly MT of Assem02.

its hash value and value2 is equal to the hash value of the string that is sequentially connected by all the children's hashes.

When a user can access all the files of Assem6 with privilege and submits the Id Assem6-1 to the CSP, the contents of the correct hash value file produced by the cloud should be: [Assen6-1, H4, hash(H1 + H2)]; [Part004-1, H1]; and [Part003-1, H2], as shown in Fig. 3(c). The user will compute the hashes of the downloaded files and compare them with the received hashes. If the files have been modified or the structure has been destroyed, these problems can easily be found by the user.

4.2.1. Deformation-based protection

Typical CAD models consist of various features $\{F_i\}$ (a set of model features), as shown in Fig. 4(a). Features are created based on a series of constraints and sketches $\{S_i\}$ (a set of sketches of model features), both of which determine the model shapes. If the deformation method is applied to both the features and the benchmarks, the validity of the CAD models will be affected. We adopt a sketch-based deformation method to hide the confidential information of the CAD models. A sketch S_i is composed of sketch elements and constraint elements. Each sketch S_i has sketch points denoted as $(x_{i,1}, x_{i,2})$ or $(x_{i,1}, x_{i,2}, x_{i,3})$ and will be synthesized into a sketch matrix $\mathbf{S}_\mathrm{M}$.

We use a transformation matrix $\mathbf{T}_\mathrm{M}$ to realize parametric deformation. After choosing the parameters of matrix $\mathbf{T}_\mathrm{M}$, we gradually increase the value of n from 0 until the sketches' deformation by $\mathbf{T}_\mathrm{M}$ will not destroy the model constraints. A new sketch matrix, $\mathbf{S}_{\mathrm{M}'}$, is obtained by multiplying $\mathbf{S}_\mathrm{M}$ by $\mathbf{T}_\mathrm{M}$, as shown in Eq. (1).

This deformation method is robust and flexible for CAD models, because sketches can be restored to the original shapes according to the encryption matrix $\mathbf{T}_\mathrm{M}$, and CAD models can also be easily restored to their original states.

$$\underbrace{\begin{bmatrix} x_{1,1} & x_{1,2} \\ \vdots & \vdots \\ x_{n,1} & x_{n,2} \end{bmatrix}}_{\mathbf{S}_\mathrm{M}} \times \mathbf{T}_\mathrm{M} = \underbrace{\begin{bmatrix} x_{1,1'} & x_{1,2'} \\ \vdots & \vdots \\ x_{n,1'} & x_{n,2'} \end{bmatrix}}_{\mathbf{S}_{\mathrm{M}'}} \tag{1}$$

The transformation matrixes for two-dimensional (2D) and 3D features are defined as follows:

$$\mathbf{T}_{\mathrm{M}2\times2} = \begin{bmatrix} a_{1,1}\cdot\eta^n & a_{1,2}\cdot\lambda^n \\ a_{2,1}\cdot\lambda^n & a_{2,2}\cdot\eta^n \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{M}3\times3} = \begin{bmatrix} a_{1,1}\cdot\eta^n & a_{1,2}\cdot\lambda^n & a_{1,3}\cdot\lambda^n \\ a_{2,1}\cdot\lambda^n & a_{2,2}\cdot\eta^n & a_{2,3}\cdot\lambda^n \\ a_{3,1}\cdot\lambda^n & a_{3,2}\cdot\lambda^n & a_{3,3}\cdot\eta^n \end{bmatrix}$$

where $a_{i,j}$ means a decimal randomly selected between 0.9 and 1.1, η and λ are two coefficient satisfied $0 < \eta$, $\lambda < 1$, and n is an integer increasing from 0. The deformation is explained in Fig. 4. As shown in Fig. 4(a), Cyt-Extrude1 is a private feature that cannot be shared with others. We modify its Sketch4 with the transformation matrix to obtain a deformed Part003, as shown in Fig. 4(c).

4.2.2. The CAD-CP-ABE scheme

The proposed scheme for CKs management is based on the bilinear maps definition and a CP-ABE encryption policy. It consists of four functions: Setup, KeyGen, Encrypt, and Decrypt.

(1) (PK, MSK) ← Setup(1^k). This function inputs a security parameter k and a prime number p, and outputs public key (PK) and master SK (MSK).

(2) (SK) ← KeyGen(PK, MSK, S). This function inputs PK, MSK, and a set of attributes of one user S, and generates a SK for the attributes set S.

(3) (ACT) ← Encrypt(PK, CK, AT). This function inputs PK, the CKs, and a set of all attributes AT, and outputs ACT for CKs.

(4) $(\mathrm{ck}_i\ (i \in [1, \mathrm{num}])) \leftarrow \mathrm{Derypt}(\mathrm{PK},\ \mathrm{ACT},\ \mathrm{SK})$. This function inputs PK, an ACT file, and the SK file of the user. SK is described by S. If S satisfies the access structure of the AHAT, some or all of the CKs, ck_i $(i \in [1, \mathrm{num}])$, can be decrypted. Then, the corresponding files m_i $(i \in [1, \mathrm{num}])$ will be decrypted with the corresponding CKs, ck_i, where num represents the number of all the CKs.

4.3. System secure co-design process

The secure co-design process comprises the following steps:

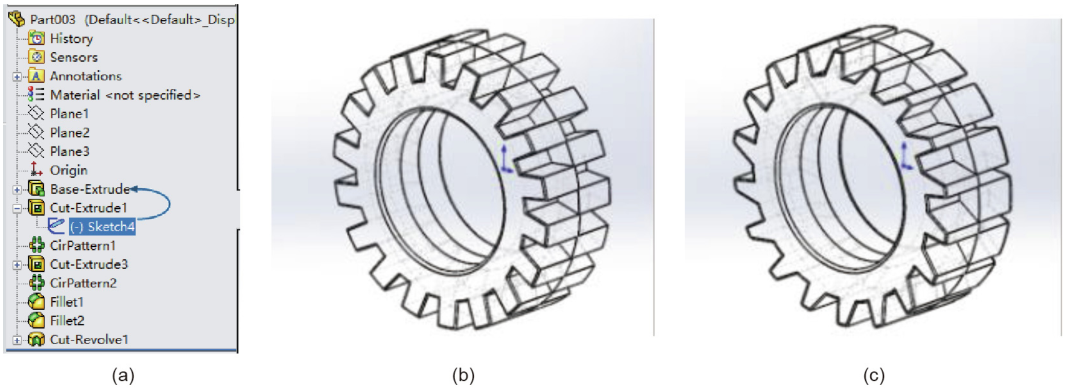
(1) **Deform private features**: For a CAD assembly model M_0 owned by the data owner (D), the private shape features (F_i) of the components inside the assembly will be hidden using a deformation-based method. M_0 is transferred into a new CAD assembly model M before sharing.

(2) **Construct an assembly MT**: The D will build an assembly MT, as shown in Fig. 3.

(3) **Enact the encryption process**: The files (i.e., plaintext) of M will be encrypted as model ciphertext $M' = \{m'_1, \ldots, m'_i, \ldots\}$ with $\mathrm{CK} = \{\mathrm{ck}_1, \ldots, \mathrm{ck}_i, \ldots\}$ by means of a symmetric encryption algorithm. The D will first generate an AHAT and then compute the ACT file for $\mathrm{CK} = \{\mathrm{ck}_1, \ldots, \mathrm{ck}_i, \ldots\}$ through the Encrypt function in the CAD-CP-ABE scheme (i.e., the CAD-CP-ABE encryption process).

(4) **Share data files**: The D will upload the assembly MT, the M', and the CKs ACT file to the CSP. The assembly MT is stored separately on cloud server A. M' and ACT are stored on cloud server B.

(5) **Verify structural integrity**: The co-design user (U) can send a set of files to the CSP, which stores the assembly MT in the cloud



(a) (b) (c)

Fig. 4. An example of feature sketch deformation for Part003. (a) The FeatureManager of Part003 in SolidWorks; (b) the original Part003; (c) the deformed Part003.

server with other shared files. This server will return a set of hash values for integrity verification.

(6) **Obtain plaintext**: The U will send its identity attributes to the trusted authority to retrieve the SKs. Ciphertext elements in the ACT file are combined with SKs to evaluate the CK. Then the encrypted model files M' are decrypted with the CKs.

5. Detailed process of the CAD-CP-ABE scheme

5.1. Hierarchy access structure construction for the CAD-AP-ABE scheme

The access structure T1 in Fig. 5(a) is based on a CP-ABE scheme to decrypt the source file m1. Each non-leaf node is a threshold node, and each leaf node is associated with attributes. The root node represents a source file. The threshold value is num1/num2, where num1 represents "the number of nodes that need to satisfy conditions" and num2 represents "the total number of children."

For a CAD assembly model structure, as shown in Fig. 5(b), nodes on a higher level would be accessed by fewer users. If attribute nodes are inserted into the model structure, repeated attribute nodes will increase the unnecessary overhead, as shown in Fig. 5(b). At the same time, a duplicate component node will also add overhead.

Therefore, we specify the following rules to generate the AHAT in our scheme:

(1) Extract the integrated assembly model structure for AHAT generation.

(2) Divide the structure into component nodes and hidden nodes. As node 2 and node 3 represent the same attribute, if they have the same parent node 1, or if the parent of node 2 is the ancestor of node 3, set node 2 and its descendant nodes to be hidden nodes in the AHAT.

(3) Insert attribute nodes and/or threshold nodes for each component from the assembly to construct the AHAT. In order to decrease the complexity of the AHAT, it is necessary to prune the attribute structures. If any component node has the same access attribute structures as its parent/ancestor nodes, these attributes are cut off and the access structure is changed.

The identity attributes are defined hierarchically, as follows: chief engineer, deputy chief engineer, project engineer, engineer, and assistant engineer. The corresponding attributes collection $\mathrm{AT} = \{1, 2, 3, 4, 5\}$. For example, we list the three companies C1, C2, and C3 as company attributes and define their values as 6, 7, and 8, respectively. The total attributes set $\mathrm{AT} = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

As shown in Fig. 6, there is an AHAT that describes an access structure and contains several access levels. The terms and functions of the AHAT and nodes are as follows:

(x, y): This binary denotes a non-hidden node in AHAT. The symbol x represents the node's row (from the top to bottom), and y represents the node's column (from left to right). For example, the binary of node Assem02 is (1, 1), the binary of node Part005-1 is (2, 1), and the binary of attribute 3 on level three is (3, 5) in Fig. 6.

$\mathrm{num}_{(x,y)}$: This denotes the number of non-hidden nodes in the children set of (x, y) in the AHAT. For example, $\mathrm{num}_{(2,2)} = 2$ in Fig. 6.

$k_{(x,y)}$: This denotes the threshold value of a non-hidden node (x, y), where $0 < k_{(x,y)} \leq \mathrm{num}_{(x,y)}$. If ($x$, y) is a leaf node, $k_{(x,y)} = 1$. For example, $k_{(1,1)} = k_{(2,2)} = 2$.

parent(x, y): This represents the parent of node (x, y) in the AHAT. For example, parent(5, 1) = (4, 3) in Fig. 6.

att(x, y): This denotes an attribute associated with the leaf node (x, y) in the AHAT.

index(x, y): This returns a unique value associated with the node (x, y). The index values are uniquely assigned to the nodes in the AHAT in an arbitrary manner for a given key.

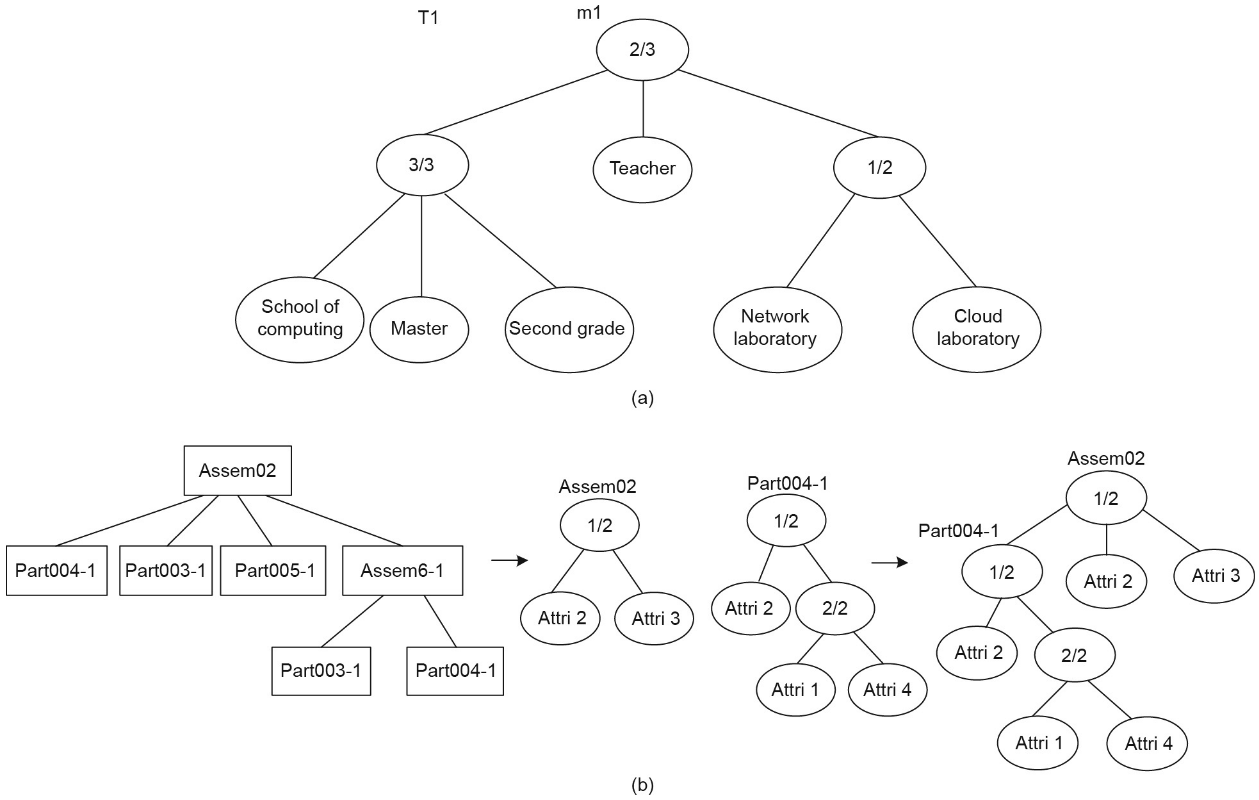


Fig. 5. An example of access structures in the CP-ABE scheme. (a) An example of an access structure tree. T1 is the access structure of m1. (b) The hierarchical access structure for Assem02 and Part004. Attri: attribute node.

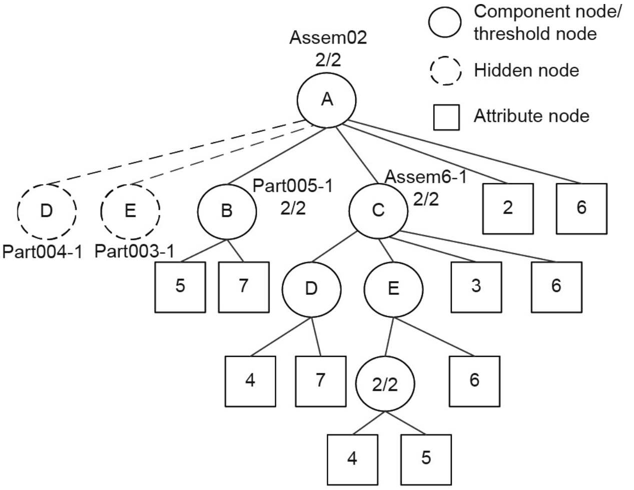


Fig. 6. The AHAT that is the integrated hierarchical access structure of the assembly model.

5.2. Encryption and decryption process of CKs

DBDH is a bilinear parameter generator. Assume that the data owner shares files with num responding CKs, $\mathrm{CK}=\{\mathrm{ck}_1,\ldots,\mathrm{ck}_{\mathrm{num}}\}$. In addition, two hash functions H_1:{0,1}* → G_0 and H_2:{0,1}* → G_T are used in the CAD-CP-ABE scheme.

Setup(1^k). The authority executes this function within a security parameter k and random numbers $\alpha,\beta\in Z_p$. The DBDH will output PK and MSK as shown in Eqs. (2) and (3), respectively.

$$\mathrm{PK}=\left\{G_0,g,h=g^{\beta},e(g,g)^{\alpha}\right\} \tag{2}$$

$$\mathrm{MSK}=\{g^{\alpha},\beta\} \tag{3}$$

KeyGen(PK, MSK, S). The authority executes this function with a set of attributes of one user S, and creates SK as shown in Eq. (4), where $r\in Z_p$ and $r_j\in Z_p$ are randomly chosen for this user. D is a normal key parameter, D_j and D_j' are key parameters that belong to attribute j. D, D_j, and D_j' make up SK.

$$\mathrm{SK}=\left\{D=g^{(\alpha+r)/\beta},\forall j\in S:D_j=g^r\cdot H_1(j)^{r_j},D_j'=g^{r_j}\right\} \tag{4}$$

In the proposed approach, the attribute set of one user has two elements: a position attribute and a company attribute. Thus, S is usually two tuples.

Encrypt(PK, CK, A). The data owner selects num random numbers $\{s_1,\ldots,s_{\mathrm{num}}\}\in Z_p$ for $\mathrm{CK}=\{\mathrm{ck}_1,\ldots,\mathrm{ck}_{\mathrm{num}}\}$, and computes C_i and C_i' for all component nodes $(i=1,2,\ldots,\mathrm{num})$, as shown in Eq. (5). C_i and C_i' are key parameters that belong to ck_i.

$$C_i=\mathrm{ck}_i\cdot e(g,g)^{\alpha s_i},\ \ C_i'=h^{s_i} \tag{5}$$

Generate a polynomial $q_{(x,y)}$ for each non-hidden node with polynomial rules in a top-down manner, as follows:

(1) Start from the root node.

(2) The degree of $q_{(x,y)}$ is $k_{(x,y)}$ – 1.

(3) If (x, y) is a component node, $q_{(x,y)}(0)$ = s_i. Otherwise, $q_{(x,y)}(0)$ = $q_{\mathrm{parent}(x,y)}$(index(x, y)). Other polynomial information of $q_{(x,y)}$ is randomly selected.

For each leaf node, the data owner computes $C^1_{(x,y)}$ and $C^2_{(x,y)}$, as shown in Eq. (6). $C^1_{(x,y)}$ and $C^2_{(x,y)}$are two key parameters used for decryption for node (x, y).

$$C^1_{(x,y)}=g^{q_{(x,y)}(0)},\ \ C^2_{(x,y)}=H_1(\mathrm{att}(x,y))^{q_{(x,y)}(0)} \tag{6}$$

For each component node that has component nodes as children, the data owner computes $C_{(x,y),l}$, as shown in Eq. (7), where the component children set is {child$_1$, …, child$_l$, …}. $C_{(x,y),l}$ is a key parameter used between parent node and children nodes.

$$C_{(x,y),l}=e(g,g)^{\alpha\cdot q_{\mathrm{child}_l}(0)}\cdot H_2\left(e(g,C'_{\mathrm{child}_l})\cdot e(g,g)^{\alpha\cdot q_{(x,y)}(0)}\right) \tag{7}$$

The data owner outputs the integrated ACT file, as shown in Eq. (8).

$$\mathrm{ACT}=\left\{\mathrm{AHAT},C_i,C_i',C^1_{(x,y)},C^2_{(x,y)},C_{(x,y),l}\right\} \tag{8}$$

Decrypt(PK, ACT, SK). A user needs the PK and the SK described by S to decrypt ACT.

For the leaf node (x, y), we define DecryptNode(ACT, SK, (x, y)) as Eq. (9), where j = att(x, y), and if $j\notin S$, DecryptNode(ACT, SK, (x, y)) = null.

$$\begin{aligned}\mathrm{DecryptNode}(\mathrm{ACT},\mathrm{SK},(x,y))&=\frac{e\left(D_j,C^1_{(x,y)}\right)}{e\left(D_j',C^2_{(x,y)}\right)}\\&=\frac{e\left(g^r\cdot H_1(j)^{r_j},g^{q_{(x,y)}(0)}\right)}{e\left(g^{r_j},H_1(\mathrm{att}(x,y))^{q_{(x,y)}(0)}\right)}\\&=e(g,g)^{rq_{(x,y)}(0)}\end{aligned} \tag{9}$$

For each component/threshold node, we define DecryptNode (ACT, SK, (x, y)) as shown in Eq. (10), where z are the attribute/threshold children of (x, y), $S_{(x,y)}$ is an arbitrary $k_{(x,y)}$-sized attribute/threshold children set of (x, y) in the AHAT, $S'_{(x,y)}=\{\mathrm{index}(z):z\in S_{(x,y)}\}$, and $\mathrm{val}=\mathrm{index}(z)$.

$$\begin{aligned}F_{(x,y)}&=\prod_{z\in S'_{(x,y)}}F_z^{\Delta\mathrm{val}S'_{(x,y)}(0)}=\prod_{z\in S'_{(x,y)}}\left(e(g,g)^{rq_z(0)}\right)^{\Delta\mathrm{val}S'_{(x,y)}(0)}\\&=\prod_{z\in S'_{(x,y)}}\left(e(g,g)^{rq_{(x,y)}(\mathrm{val})}\right)^{\Delta\mathrm{val}S'_{(x,y)}(0)}=\prod_{z\in S'_{(x,y)}}e(g,g)^{rq_{(x,y)}(0)}\end{aligned} \tag{10}$$

Because the random number s_i of component (x, y) is irrelevant with its parent's random number, z cannot be the component node.

Next, $e(g,g)^{\alpha s_i}$ can be computed by means of Eq. (11), where i is the number of the component node (x, y).

$$F'_{(x,y)}=\frac{e(C_i',D)}{F_{(x,y)}}=\frac{e\left(h^{s_i},g^{(\alpha+r)/\beta}\right)}{e(g,g)^{rq_{(x,y)}(0)}}=e(g,g)^{\alpha s_i},\ \ i\in[1,\mathrm{num}] \tag{11}$$

Because the parent file needs to use all the children files, (x, y) has access to decrypt all the children files. The $F_{(x,y),l}$ is the intermediate parameter of the decryption. We can compute $F_{(x,y),l}$ of component child l for component node (x, y) in the AHAT using Eq. (12).

$$F_{(x,y),l}=\frac{C_{(x,y),l}}{H_2\left[e(g,C_l')\cdot F_{(x,y)}\right]}=e(g,g)^{\alpha q_{\mathrm{child}_l}(0)},\ \ l=1,2,\ldots \tag{12}$$

Then, the corresponding CKs, ck$_i$, are decrypted by executing Eq. (13), where i is the number of the component node (x, y).

$$\mathrm{ck}_i=\frac{C_i}{F'_{(x,y)}}=\frac{\mathrm{ck}_i\cdot e(g,g)^{\alpha s_i}}{e(g,g)^{\alpha s_i}},\ \ i\in[1,\mathrm{num}] \tag{13}$$

Finally, ck$_i$ is used to decrypt the corresponding file.

5.3. Security proof for the CAD-CP-ABE scheme

5.3.1. Security model

In the proposed scheme, the SK for the user is associated with an attribute set, and the ACT is associated with the access structure. The security model of our scheme should resist the CPA. The CPA security game between the adversary, A1, and the challenger, B1, requires that A1 select the challenging structure AT* and can require all SK where SK does not satisfy AT*.

(1) **Initialization**. A1 selects a challenging structure AT* and delivers it to B1.

(2) **Setup**. B1 runs the Setup(1^k) algorithm and sends PK to A1.

(3) **Query Phase 1**. A1 selects a series of attribute sets $S_1, ..., S_w, \ \forall i \in [1, w], S_i \notin \mathrm{AT}^*$ to repeatedly query B1 for the SK. B1 answers these queries by running the KeyGen(PK, MSK, S_i) algorithm.

(4) **Challenge**. A1 selects two messages, m_0 and m_1, which are of equal length, to be challenged. Then B1 randomly selects a bit $\mu \in \{0, 1\}$ and encrypts m_μ with access structure AT*. Finally, B1 gives the ciphertext ACT* to A1.

(5) **Query Phase 2**. This is the same as Query Phase 1.

(6) **Guess**. A1 outputs a guess bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, A1 wins the security game; otherwise, it fails. The advantage of A1 in winning the CPA game is defined as $\mathrm{Adv}_{\mathrm{A1}}^{\mathrm{CPA}}(1^k) = |\Pr[\mu' = \mu] - 1/2|$, where Pr stands for probability.

Definition 3. The CAD-CP-ABE scheme is secure against the CPA if no probabilistic polynomial-time adversary A1 can win the security game.

5.3.2. Security proof for the proposed scheme

Theorem 1: Suppose the DBDH assumption holds in $< G_0, G_\mathrm{T} >$, then no polynomial adversary can selectively break the proposed scheme.

Proof: Assume that adversary A1 comes against our construction with a non-negligible advantage $\varepsilon = \mathrm{Adv}_{\mathrm{A1}}^{\mathrm{CPA}}(1^k)$ in the selective security game. Challenge B1 can distinguish the DBDH tuple D_bdh and the random tuple D_rand with a non-negligible probability, $\varepsilon/2$. Let $e : G_0 \times G_0 \to G_\mathrm{T}$ be an efficiently computable bilinear map, where G_0 has prime order p with generator g. The challenger randomly selects parameters $(a, b, c) \in Z_p$, a random value $u \in \{0, 1\}$, and a random element $\theta \in_R G_\mathrm{T}$. If $u = 0$, then the challenger B1 sets $(g, A, B, C, T) = (g, g^a, g^b, g^c, e(g, g)^{abc}) \in D_\mathrm{bdh}$; otherwise, the challenger sets $(g, A, B, C, T) = (g, g^a, g^b, g^c, \theta) \in D_\mathrm{rand}$. The ACT can be calculated as $\mathrm{ACT} = \left\{\mathrm{AHAT}, C_i, C_i', C^1_{(x,y)}, C^2_{(x,y)}, C_{(x,y),l}\right\}$, according to Eq. (8).

(1) **Initialization**. Adversary A1 selects the challenging structure AT* and sends AT* to the challenger, B1.

(2) **Setup**. To provide a PK to A1, B1 randomly chooses a number $a' \in Z_p$, and defines $\partial = a' + ab$. It computes $e(g, g)^\partial = e(g, g)^{a'} \cdot e(g, g)^{ab}$. Meanwhile, it sets $h = g^\beta = g^b$. For given β, $g^\beta = g^b$. Finally, B1 gives PK to A1.

(3) **Query Phase 1**. In this phase, A1 can query the SK by submitting an attribute set $W_j = \{a_j, a_j \in \mathrm{AT}\}(a_j \notin \mathrm{AT}^*)$ to B1. Later, B1 randomly picks a number $r' \in Z_p$ and sets $r' = r - a$. B1 can obtain $D = g^{(\partial + r')/\beta} = g^{(\partial + r - a)/\beta}$. Then, for each attribute $a_j \in W_j$, B1 needs to randomly choose $r_j \in Z_p$. It constructs the remaining SK as follows: $D_j = g^{(r-a)} \cdot H_1(j)^{r_j}, \ D_j' = g^{r_j}$. Finally, B1 sends SK to A1.

(4) **Challenge**. A1 submits two messages, mess_0 and mess_1, of equal length to B1. B1 randomly generates a bit, $\mu \in \{0, 1\}$. With the encryption operation under AT*, B1 computes ACT* as $C' = h^s = g^{\beta c}, C = m_\mu \cdot e(g, g)^{\partial s} = m_\mu \cdot e(g, g)^{(a' + ab)c}$. Finally, B1 sends ACT* to A1.

(5) **Query Phase 2**. This is the same as Query Phase 1.

(6) **Guess.** Finally, A1 outputs a guess bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, B1 outputs 0 to indicate $(g, A, B, C, T) \in D_\mathrm{bdh}$. Otherwise, B1 outputs 1 to indicate $(g, A, B, C, T) \in D_\mathrm{rand}$. The probability of success of adversary A1 in the game with challenger B1 is calculated as follows.

If $(g, g^a, g^b, g^c, T) \in D_\mathrm{bdh}$, that is, $T = g^{abc}$, then ACT* is a valid ciphertext; in this case, the advantage of adversary A1 is ε.

$$\Pr[\mathrm{B1}(g, A, B, C, T) \in D_\mathrm{bdh} = 0] = 1/2 + \varepsilon$$

If $(g, g^a, g^b, g^c, T) \in D_\mathrm{rand}$, the inequation $\mu' \neq \mu$ holds. With an advantage of 1/2, adversary A1 has nothing to do with the distribution on μ'. In this case, there is no advantage for adversary A1.

$$\Pr[\mathrm{B1}(g, A, B, C, T) \in D_\mathrm{rand} = 0] = 1/2$$

Lastly, the advantage of the challenge B1 is described as follows:

$$\begin{aligned}\mathrm{Adv}_{\mathrm{B1}} &= 1/2\{\Pr[\mathrm{B1}(g, A, B, C, T) \in D_\mathrm{bdh} = 0] + \Pr[\mathrm{B1}(g, A, B, C, T) \in D_\mathrm{rand} = 0]\} - 1/2 \\ &= 1/2(1/2 + \varepsilon + \ 1/2) - 1/2 = \varepsilon/2\end{aligned}$$

5.4. Theoretical analysis

Let C_e be the e operation (bilinear pairing). Suppose that $|A_\mathrm{a}|$ is the number of attribute nodes, A_c is the set of component nodes that have component nodes as children, and A_u is the attributes set of user U.

There are k CKs, and each node in A_c contains n component nodes as children. In the CAD-CP-ABE scheme, some attribute nodes of lower level component nodes are cut off in generating the AHAT. Thus, when the number of CKs is fixed, the encryption time is related to k, $n|A_\mathrm{c}|$, and $|A_\mathrm{a}|$.

In our approach, $|A_\mathrm{u} = 2|$ is constant, which makes the decryption time independent of the leaf nodes. Suppose user U has privileges to access the root node. Because the minimum number of interior nodes satisfying a root access structure is two, there are just two attribute nodes. The root node is calculated as shown in Eq. (9). The decryption time is related to k and $n|A_\mathrm{c}|$.

Moreover, the size of the ACT can be obtained by Eq. (8), as shown in Table 1, where L is the length of elements $G_i, i \in \{0, \mathrm{T}\}$.

6. Experiments

6.1. Experimental simulation

This approach executes two encryption processes: plaintext encryption and CKs encryption. For plaintext encryption, we use the advanced encryption system (AES) algorithm to encrypt and decrypt the assembly files. For CKs encryption, we implement the CAD-CP-ABE scheme of the CKs based on a Java pairing-based cryptography (JPBC) library [54]. We use a Type A bilinear map. Type A pairings are constructed on the curve $y^2 = x^3 + x$ over the field of F_q. This pairing is symmetrical, and the order r is some prime factor of q + 1. Type A needs two parameters rBits = 160. All results are the averages of 20 experiments.

6.2. Experimental results

As shown in Fig. 7, we use the AES algorithm to encrypt and transform an assembly format to a text format, and the decryption process is the reverse procedure of encryption. This experiment confirms the practicability of an encryption algorithm on the assembly files, where the encryption and decryption processes will not destroy the content integrity.

As shown in Fig. 8(a), the generation time of SKs is approximately linear. Because some parameters of each experiment are newly selected, a slight computation error is caused in a linear relationship. We use method newElement() for Object Pairing to generate a SK. The time of this process is less than 2 s, and all

Table 1
Features of CAD-CP-ABE ($\mathrm{CK} = \{\mathrm{ck}_1, ..., \mathrm{ck}_k\}$).

Feature	Computing equation
Encryption time	$(2\lvert A_\mathrm{a}\rvert + k)G_0 + 2(n\lvert A_\mathrm{c}\rvert + k)G_\mathrm{T}$
Decryption time	$5C_e + (4 + n\lvert A_\mathrm{c}\rvert + 2k)G_\mathrm{T}$
The size of ACT	$(2\lvert A_\mathrm{a}\rvert + k)L_{G_0} + (n\lvert A_\mathrm{c}\rvert + k)L_{G_\mathrm{T}}$

processes are not greater than 50 s. We choose to use the method newRandomElement( ) to strengthen security in this process, which results in extra time cost.

The encryption and decryption experimental results are presented in Figs. 8(b)–(d). As shown in Fig. 8(b), we assume that there are various leaf nodes with two hierarchy files. Based on the rules of AHAT generation, the number of leaf nodes in CAD-CP-ABE is equal to $|A_{a2}|$ in CP-ABE. As shown in Fig. 8(c), there are various hierarchy files with fixed leaf nodes (N = 30). Figs. 8(b) and (c) show that the results increasingly and approximately follow a linear relationship. Obviously, the number of leaf nodes has a greater impact on the time cost in CAD-CP-ABE.

The decryption time of the CKs is counted from the root node. In our approach, just two leaf nodes need to be satisfied for the root node in CAD-CP-ABE and for each component node in CP-ABE. When a user U obtains its CK for file A, then U can derive CKs for all other component nodes under A layer by layer within the ACT file. The decryption time cost is only related to the number CKs, as shown in Fig. 8(d).

The experiments demonstrate that the proposed scheme improves the encryption and decryption efficiency.

7. Conclusion

In this paper, we proposed a hierarchical assembly files-sharing approach to protect CAD models for outsourcing and co-design in the era of cloud-based design and manufacture. This approach combines an ABE scheme, structural integrity checking, and deformation-based shape protection for CAD assembly models. The simulation experiments demonstrated that the approach is feasible in terms of computation efficiency and flexibility.

For future works, the first direction is to determine how to extend the shape encryption from sketch-based CAD part deformation to CAD assembly deformation. The second direction is to adopt multi-core computing techniques and optimization approaches [55–61] in order to accelerate the encryption and decryption for CAD big data. The third direction is to extend the proposed approach to other multimedia data [62–67]. Finally, we will

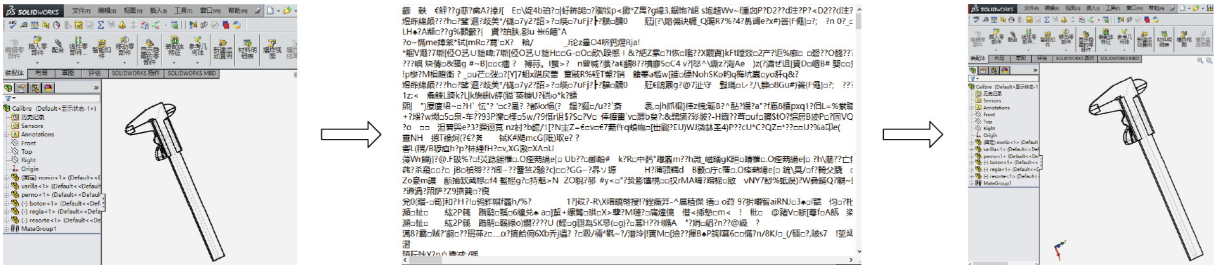
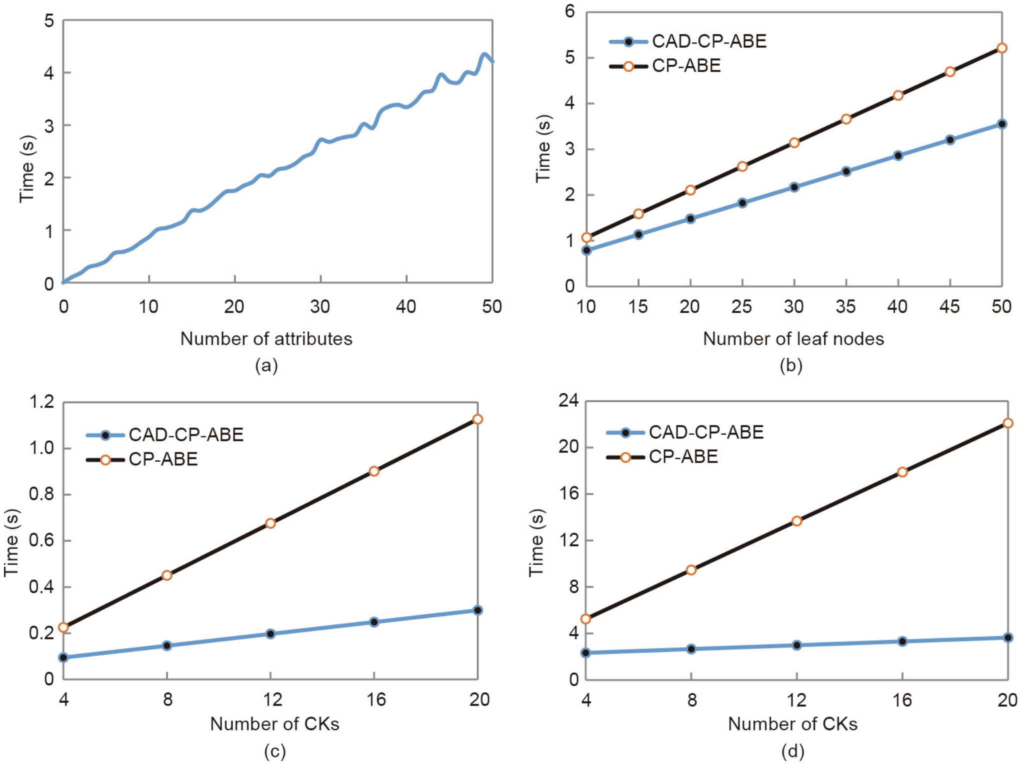


Fig. 7. An example of assembly file encryption and decryption with the AES algorithm.



(a)

(b)

(c)

(d)

Fig. 8. Experimental results of CKs encryption and decryption. (a) SKs generation time and all number of attributes; (b) encryption time cost for two CKs; (c) encryption time cost for 30 leaf nodes; (d) decryption time cost of all CKs from the root node.

integrate a searchable encryption scheme into the proposed approach [68,69].

Acknowledgments

The work is supported by the National Natural Science Foundation of China (62072348) and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies; 2019AEA170).

Compliance with ethics guidelines

Yueting Yang, Fazhi He, Soonhung Han, Yaqian Liang, and Yuan Cheng declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Zhong RY, Xu X, Klotz E, Newman ST. Intelligent manufacturing in the context of industry 4.0: a review. Engineering 2017;3(5):616–30.

[2] Wang L, Chen X, Liu Q. A lightweight intelligent manufacturing system based on cloud computing for plate production. Mob Netw Appl 2017;22(6):1170–81.

[3] Andreadis G, Fourtounis G, Bouzakis KD. Collaborative design in the era of cloud computing. Adv Eng Softw 2015;81:66–72.

[4] Tao F, Qi Q, Wang L, Nee A. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: correlation and comparison. Engineering 2019;5(4):653–61.

[5] Chen Y. Integrated and intelligent manufacturing: perspectives and enablers. Engineering 2017;3(5):588–95.

[6] Wu D, Rosen DW, Wang L, Schaefer D. Cloud-based design and manufacturing a new paradigm in digital manufacturing and design innovation. Comput Aided Des 2015;59:1–14.

[7] Cai W, He F, Lv X, Cheng Y. A semi-transparent selective undo algorithm for multi-user collaborative editor. Front Comput Sci 2021;25(3):1–21.

[8] Wang J, Zheng P, Lv Y, Bao J, Zhang J. Fog-IBDIS: industrial big data integration and sharing with fog computing for manufacturing systems. Engineering 2019;5(4):662–70.

[9] Villa A, Taurino T. From industrial districts to SME collaboration frames. Int J Prod Res 2018;56(1–2):974–82.

[10] Tehrani SR, Shirazi F. Factors influencing the adoption of cloud computing by small and medium size enterprises (SMEs). In: Proceeding of International Conference on Human Interface and the Management of Information; 2014 Jun 22–27; Heraklion, Greece. Berlin: Springer; 2014. p. 631–42.

[11] Cheng Y, Bi L, Tao F, Ji P. Hypernetwork-based manufacturing service scheduling for distributed and collaborative manufacturing operations towards smart manufacturing. J Intell Manuf 2020;31(7):1707–20.

[12] Zhang Y, Xi D, Yang H, Tao F, Wang Z. Cloud manufacturing based service encapsulation and optimal configuration method for injection molding machine. J Intell Manuf 2019;30(7):2681–99.

[13] Zhao C, Zhang L, Ren L, Tao F. Simulation platform for transaction processes in cloud manufacturing. Comput Integr Manuf Syst 2016;22(1):25–32.

[14] Demoly F, Roth S. Knowledge-based parametric CAD models of configurable biomechanical structures using geometric skeletons. Comput Ind 2017;92–93:104–17.

[15] Qin F, Gao S, Yang X, Li M, Bai J. An ontology-based semantic retrieval approach for heterogeneous 3D CAD models. Adv Eng Inform 2016;30(4):751–68.

[16] Liang Y, He FX, Zeng X. 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. Integr Comput Aided Eng 2020;27(4):417–35.

[17] Shen W, Li W. Collaboration computing technologies and applications. J Netw Comput Appl 2013;36(6):1577–8.

[18] Li W, Shen W. Collaborative design: new methodologies and technologies. Comput Ind 2008;59:853–4.

[19] Gong W, Wang QS, Chen HQ. Summarization on intelligent manufacturing information security certification feasibility research. Inf Technol Inf 2018;2–3:147–50.

[20] Chang SI, Chang IC, Li HJ, He TH. The study of intelligent manufacturing internal control mechanism by using a perspective of the production cycle. J Ind Prod Eng 2014;31(3):119–27.

[21] Kim H, Yeo C, Lee ID, Mun D. Deep-learning-based retrieval of piping component catalogs for plant 3D cad model reconstruction. Comput Ind 2020;123:103320.

[22] Liu Y, Wang L, Wang XV, Xu X, Zhang L. Scheduling in cloud manufacturing: state-of-the-art and research challenges. Int J Prod Res 2019;57(15–16):4854–79.

[23] Liu Y, Wang L, Wang XV, Xu X, Jiang P. Cloud manufacturing: key issues and future perspectives. Int J Comput Integr Manuf 2019;32(9):858–74.

[24] Sahai A, Waters B. Fuzzy identity-based encryption. In: Proceeding of Annual International Conference on the Theory and Applications of Cryptographic Techniques; 2005 May 22–26; Aarhus, Denmark. Berlin: Springer; 2005. p. 457–73.

[25] Miao Y, Ma J, Liu X, Li X, Jiang Q, Zhang J. Attribute-based keyword search over hierarchical data in cloud computing. IEEE Trans Serv Comput 2020;13(6):985–96.

[26] Wan Z, Liu J, Deng RH. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans Inf Forensic Secur 2012;7(2):743–54.

[27] Wang S, Zhou J, Liu JK, Yu J, Chen J, Xie W. An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans Inf Forensic Secur 2016;11(6):1265–77.

[28] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceeding of 2007 IEEE Symposium on Security and Privacy (SP '07); 2007 May 20–23; Berkeley, CA, USA. New York: IEEE; 2007. p. 321–34.

[29] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. Computer 1996;29(2):38–47.

[30] Oh S, Park S. Task–role-based access control model. Inf Syst 2003;28(6):533–62.

[31] Park J, Sandhu R. Towards usage control models: beyond traditional access control. In: Proceeding of the Seventh ACM Symposium on Access Control Models and Technologies; 2002 Jun; Monterey, CA, USA. New York: Association for Computing Machinery; 2002. p. 57–64.

[32] Park J, Sandhu R. The UCON$_{\mathrm{ABC}}$ usage control model. ACM Trans Inf Syst Secur 2004;7(1):128–74.

[33] Lampson BW. Protection. Oper Syst Rev 1974;8(1):18–24.

[34] Cera CD, Kim T, Han J, Regli WC. Role-based viewing envelopes for information protection in collaborative modeling. Comput Aided Des 2004;36(9):873–86.

[35] Yao L, Shao J, Sheng G, Zhang G. Research on a security model of data in computer supported collaborative design integrated with PDM system. In: Proceeding of Workshop on Intelligent Information Technology Application (IITA 2007); 2007 Dec 2–3; Zhangjiajie, China. New York: IEEE; 2007. p. 91–4.

[36] Chang H, Kim KK, Kim Y. The development of security system for sharing cad drawings in U-environment. Comput Inf 2008;27(5):731–41.

[37] Speier C, Whipple JM, Closs DJ, Voss MD. Global supply chain design considerations: mitigating product safety and security risks. J Oper Manag 2011;29(7–8):721–36.

[38] Zeng Y, Wang L, Deng X, Cao X, Khundker N. Secure collaboration in global design and supply chain environment: problem analysis and literature review. Comput Ind 2012;63(6):545–56.

[39] Wang Y, Ajoku PN, Brustoloni JC, Nnaji BO. Intellectual property protection in collaborative design through lean information modeling and sharing. J Comput Inf Sci Eng 2006;6(2):149–59.

[40] Cheng H, Li X. Partial encryption of compressed images and videos. IEEE Trans Signal Process 2000;48(8):2439–51.

[41] Nishchal NK, Naughton TJ. Flexible optical encryption with multiple users and multiple security levels. Opt Commun 2011;284(3):735–9.

[42] Huang Z, Liu G, Ren Z, Zeng L. A method of 3D data information encryption with virtual holography. In: Proceeding of Eighth International Symposium on Optical Storage and 2008 International Workshop on Information Data Storage; 2008 Nov 24–27; Wuhan, China. New York: International Society for Optics and Photonics; 2009. p. 71250E.

[43] Kim KC, Yoo SB. Collaborative design by sharing multiple-level encryption files. Concurrent Eng 2014;22(1):29–37.

[44] Chen T, Tsai HR. Ubiquitous manufacturing: current practices, challenges, and opportunities. Robot Comput Integr Manuf 2017;45:126–32.

[45] Cai X, He F, Li W, Li X, Wu Y. Multi-granularity partial encryption method of cad model. In: Proceeding of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2013 Jun 27–29; Whistler, BC, Canada. New York: IEEE; 2013. p. 23–30.

[46] Cai X, Li W, He F, Li X. Customized encryption of computer aided design models for collaboration in cloud manufacturing environment. J Manuf Sci Eng 2015;137(4):040905.

[47] Cai X, Wang S, Lu X, Li W. Parametric encryption of cad models in cloud manufacturing environment. In: Proceeding of the 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2016 May 4–6; Nanchang, China. New York: IEEE; 2016. p. 551–6.

[48] Cai X, Wang S, Lu X, Li W. An encryption approach for product assembly models. Adv Eng Inform 2017;33:374–87.

[49] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceeding of the 13th ACM conference on Computer and communications security; 2006 Oct; Alexandria, VA, USA. New York: Association for Computing Machinery; 2006. p. 89–98.

[50] Sun X, Zeng Y, Liu W. Formalization of design chain management using environment-based design (EBD) theory. J Intell Manuf 2013;24(3):597–612.

[51] Liu W, Zeng Y. Conceptual modeling of design chain management towards product lifecycle management. In: Chou SY, Trappey A, Pokojski J, Smith S, editors. Global perspective for competitive enterprise, economy and ecology. Berlin: Springer; 2009. p. 137–48.

[52] Zeng Y. Environment-based design (EBD): a methodology for transdisciplinary design. J Integr Des Process Sci 2015;19(1):5–24.

[53] Merkle RC, inventor; The Board of Trustees of the Leland Stanford Junior University, assignee. Method of providing digital signatures. United States patent US 4200770. 1982 Jan 5.

[54] De Caro A, Iovino V. jPBC: Java pairing based cryptography. In: Proceeding of 2011 IEEE Symposium on Computers and Communications (ISCC); 2011 Jun 28–Jul 1; Kerkyra, Greece. New York: IEEE; 2011. p. 850–5.

[55] Hou N, He F, Zhou YCY, Chen Y. An efficient GPU-based parallel tabu search algorithm for hardware/software co-design. Front Comput Sci 2020;14 (5):145316.

[56] Gao Y, Gao L, Li X, Wang XV. A multilevel information fusion-based deep leaning method for vision-based defect recognition. IEEE Trans Instrum Meas 2020;69(7):3980–91.

[57] Luo J, He F, Li H, Zeng X, Liang Y. A novel whale optimization algorithm with filtering disturbance and non-linear step. Int J Bio-inspired Comput 2021;16:1–11.

[58] Li H, He F, Chen Y, Pan Y. MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. Memet Comput 2021;13(1):1–18.

[59] Wang K, Li X, Gao L. A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. Comput Ind Eng 2019;130:634–49.

[60] Chen Y, He F, Li H, Zhang D, Wu Y. A full migration BBO algorithm with enhanced population quality bounds for multimodal biomedical image registration. Appl Soft Comput 2020;93:106335.

[61] Song W, Lai M, Li X, Song Y, Gao L. A new spectral clustering based on particle swarm optimization for unsupervised fault diagnosis of bearings. In: Proceeding of 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE); 2019 Aug 22–26; Vancouver, BC, Canada. New York: IEEE; 2019. p. 386–91.

[62] Halima I, Laferte JM, Cormier G. Depth and thermal information fusion for head tracking using particle filter in a fall detection context. Integr Comput Aided Eng 2020;27(2):195–208.

[63] Pan Y, He F, Yu H. Learning social representations with deep autoencoder for recommender system. World Wide Web 2020;23(4):2259–79.

[64] Zhang S, He F, Ren W, Yao J. Joint learning of image detail and transmission map for single image dehazing. Vis Comput 2020;36(2):305–16.

[65] Halima I, Laferte JM, Cormier G, Fougères AJ, Dillenseger JL. Depth and thermal information fusion for head tracking using particle filter in a fall detection context. Integr Comput Aided Eng 2020;27(2):195–208.

[66] Quan Q, He F, Li H. A multi-phase blending method with incremental intensity for training detection networks. Vis Comput 2021;37(2):245–59.

[67] Zhang S, He F. DRCDN: learning deep residual convolutional dehazing networks. Vis Comput 2020;36(9):1797–808.

[68] Kamara S, Papamanthou C. Parallel and dynamic searchable symmetric encryption. In: Proceeding of 2013 International Conference on Financial Cryptography and Data Security; 2013 Apr 1–5; Okinawa, Japan. Berlin: Springer; 2013. p. 258–74.

[69] Cui J, Zhou H, Zhong H, Xu Y. AKSER: attribute-based keyword search with efficient revocation in cloud computing. Inf Sci 2018;423:343–52.