future internet

MDPI

Article

A Watermark-Based In-Situ Access Control Model for Image Big Data

Jinyi Guo [1,2], Wei Ren [1,2,3,*], Yi Ren [4] and Tianqing Zhu [5]

1 School of Computer Science, China University of Geosciences, Wuhan 430074, China; jinyi_g@cug.edu.cn
2 Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences (Wuhan), Wuhan 430074, China
3 Guizhou Provincial Key Laboratory of Public Big Data, GuiZhou University, Guizhou 550025, China
4 School of Computing Science, University of East Anglia, Norwich NR4 7TJ, UK; E.Ren@uea.ac.uk
5 School of Software, University of Technology Sydney, Ultimo, Sydney, NSW 2007, Australia; Tianqing.zhu@uts.edu.au
* Correspondence: weirencs@cug.edu.cn; Tel.: +86-27-6788-3716

Received: 28 June 2018; Accepted: 26 July 2018; Published: 29 July 2018

Abstract: When large images are used for big data analysis, they impose new challenges in protecting image privacy. For example, a geographic image may consist of several sensitive areas or layers. When it is uploaded into servers, the image will be accessed by diverse subjects. Traditional access control methods regulate access privileges to a *single* image, and their access control strategies are stored in servers, which imposes two shortcomings: (1) fine-grained access control is not guaranteed for areas/layers in a single image that need to maintain secret for different roles; and (2) access control policies that are stored in servers suffers from multiple attacks (e.g., transferring attacks). In this paper, we propose a novel watermark-based access control model in which access control policies are associated with objects being accessed (called an in-situ model). The proposed model integrates access control policies as watermarks within images, without relying on the availability of servers or connecting networks. The access control for images is still maintained even though images are redistributed again to further subjects. Therefore, access control policies can be delivered together with the big data of images. Moreover, we propose a hierarchical key-role-area model for fine-grained encryption, especially for large size images such as geographic maps. The extensive analysis justifies the security and performance of the proposed model.

Keywords: access control; watermark; image; big data

1. Introduction

The development of deep learning enables the analysis of a massive amount of image data. During these processes, how to analyse the image data while protecting images from leakage and exposure is a big challenge. Traditional access control policies may be invalid when images are stored again in different servers. For example, 4G and incoming 5G techniques enable smart phone users to share their images easily. When an image is uploaded to a service provider (e.g., Facebook), a user can set access privileges to control the access rights for the image so that the image can only be accessed by "friends" or the public. However, when the accumulated images are redistributed to other parties for further analysis (e.g., the Facebook-Cambridge Analytica scandal [1]), the access control policies which were stored in its original servers are lost. Thus, desired protection solutions should integrate access control policies with the image itself. In other words, even if the image is redistributed, the access control polices will be attached (in-situ) as well. In addition, a simple "yes" or "no" access control on an image does not work well. For example, when taking a photo with a smart phone, additional information such as location data, latitude/longitude, map, date and time, etc. are also included. Various privileges

should be attached for that information. Consider geographic or remote sensing images as another example. A geographic image may consist of several areas/layers. Thus, differentiating access control for various areas/layers requires fine-grained and flexible access control policies.

Recently, centrally regulated access control models (e.g., [2,3]) have been intensively studied. However, they are not suitable for image data sharing and redistribution for the following reasons: Distributed data can be accessed with two modes: "Yes" for all or "No" for all. For data that cannot be accessed publically, the data cannot be distributed. Once data is distributed, it can be accessed by all accessors. Besides, for those data that must be in access control (classified data), control policies are difficult to define and change, especially when the data volume is large. For example, for different areas in a single image with different access policies, we must set up different regulations in central control servers. Moreover, classified data can be accessed only when remote policy conformance servers are available. The accessibility of the data relies on the availability of networks and the workload of central control servers. It constrains the convenience of remotely accessing data. Furthermore, access control regulation for a large volume of data results in a large delay. Each time accessors request images, they must first fetch access control policies on servers. In big data scenarios, accessing a response on servers results in a large burden and access delay. Finally, once data is distributed, the control domain is changed. Thus, the old management authority may not be available to control the data.

Therefore, with the development of big data sharing and redistribution, traditional access control models based on central conformance should be improved to cater to the new requirements.

In this paper, we design a novel access control model in which access control is conducted by specific clients and access policies are carried together with access objects themselves. Our proposed access control model has the following advantages: Access control policies are attached with image data. Regardless of how many times the data are further redistributed, access control policies are still incorporated with the data. Additionally, access control is fine-grained. For images with large size (e.g., geographic or remote sensing images), control strategies must be specific to different partial areas instead of the entire image. In other words, different parts in one image must conform to different access privileges. Furthermore, accessing classified data does not rely on remote servers or available network connections. The control flow is made more lightweight due to reshaping regulations at clients(we also call it in-situ control).

Based on the above observations and analysis, we propose a new access control model for big image data sharing and redistribution. The major contributions of this paper are listed as follows:

1. We propose a watermark-based access control model, allowing objects being accessed to integrate together with access control strategies.
2. We propose a hierarchical key-role-area access control model for images with large size such as geographic graphs and remote sensing graphs. We also propose a hierarchical key generation method that can guarantee fine-grained access privileges.

The rest of the paper is organized as follows: Section 2 surveys related work. Section 3 formulates the research problems and challenges. Section 4 elaborates on the proposed models. Extensive analysis of the proposed scheme is presented in Section 5, and we conclude the paper in Section 6.

2. Related Work

The topic of watermarks has been explored for decades. Due to powerful software and personal computers, there has emerged considerable unauthorized copying and distribution of digital content, such as e-books, videos, and digital images. To solve this problem, watermarks are usually used to verify and protect the copyrights [4–6]. In the above methods, both fragile watermarks and robust watermarks are coded as a legal label instead of as a control technique. Additionally, many methods have been proposed to detect the modification of images[7,8], but they are unable to find the modifier or prevent such modifications.

In recent years, several watermark schemes have been put forward for access control. Watermarks used for permitting hierarchical access control and protecting the content of visual medical information were proposed [9]. However, original images are not encrypted in this scheme. A removable and visible watermarking by combining block truncation coding and chaotic map is proposed in [10], which can be applied in copyright notification and access control in mobile communication. They proposed two-stage watermarks that blur original images before visitors pass access control, and only authorized visitors can attain clear images. However, it is not a hierarchical access control. A. Phadikar proposed a data hiding scheme for access control and error concealment in digital images [11]. He also proposed a data hiding method that integrates access control and authentication in a single platform, especially for cover images [12]. Encrypted digital images are displayed in lower quality before watermarks are read. To summarize, the schemes above display images in lower-quality formats before visitors obtain permissions. The access control strategies are still not coded in watermarks.

Quality access control is are used in audio watermarks. K. Datta et al. proposed a combination of both encryption and audio watermarking. This method is used for the safe distribution of audio content over public networks, whereby only authorized users can access the high-quality content, while other users can only access a low-quality content [13]. Watermarks can be used in video files to identify pirates, which can be extracted at the decoder and used to determine whether the video content is watermarked [14]. We stress that our proposed scheme for integrating access control policies as watermarks can also be applied in audio files or video files, although we concentrate on images in this paper.

Geologic mapping and the design of geologic (thematic) maps are currently supported by Geographic Information Systems (GIS). In order to gain a high degree of efficiency and to allow the exchange of a common structured framework, map data models have been designed by agencies and individuals in order to support their mapping process. File-based geo-databases are much more accessible, but still suffer from a number of administrative limitations [15]. A new access control mechanism that combines trust and role-based access control models is presented in [16]. J. Kim proposes a multi-layer based access control model for GIS mobile web services [17]. The objective of such spatially-aware access control models is to regulate the access to protect objects based on the position information. M. Kirkpatrick proposed role-based access control with spatial constraints [18]. F. Ma et al. proposed a fine-grained access control model for spatial data in a grid environment based on a role-based access control model [19]. Furthermore, a multi-granularity spatial access control model was proposed that introduces more types of policy rule conflicts than single-granularity objects [20]. The model can manage and enforce the strong and efficient access control technology in large-scale environments. However, all of these access control strategies are not encoded into watermarks, and access control still relies on servers.

In recent years, Quick Response (QR) codes have been popular due to their efficiency and security. They are widely used in mobile phones (e.g., applications of instance messaging, user login, and mobile payment). QR codes can not only store large information, but also have error-correction ability [21]. In addition, QR codes have high recognition rate, and there are massive algorithm libraries to invoke [22]. For these reasons, we chose the QR code as a case study for our model.

3. Problem Formulation

3.1. System Model

Figure 1 depicts the traditional access model, which includes four entities: servers, accessors, images, and access control unit. The access control unit is located with servers. Traditional access control processes include four steps, as follows: (1) Accessors request to fetch some data (e.g., images) from servers; (2) Servers inquire access control strategies from the access control unit to determine corresponding accessible objects (e.g., images); (3) The access control unit regulates access privileges as well as accessible objects accordingly; (4) Servers return accessible objects to accessors corresponding to designated privileges.

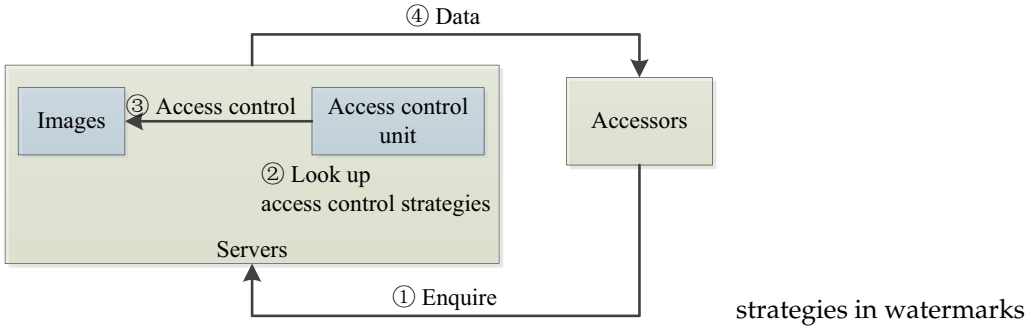

Figure 1. Existing traditional access control model.

Once accessors enquire servers for data, servers first have to search access control strategies. According to the access control policies, servers then decide what data can be provided to accessors.

In big data publication scenarios, we move the access control unit to clients, so as to provide persistent control. We change the access control processes as follows: (1) Servers incorporate access control strategies into images as watermarks. (2) Accessors request to fetch some data (e.g., images), and servers publish image big data to accessors. (3) The access control unit in clients parses access control strategies in watermarks to determine access to objects in images. (4) The access control unit regulates access privileges and returns accessible objects to accessors.

Figure 2 depicts our proposed new access control architecture.

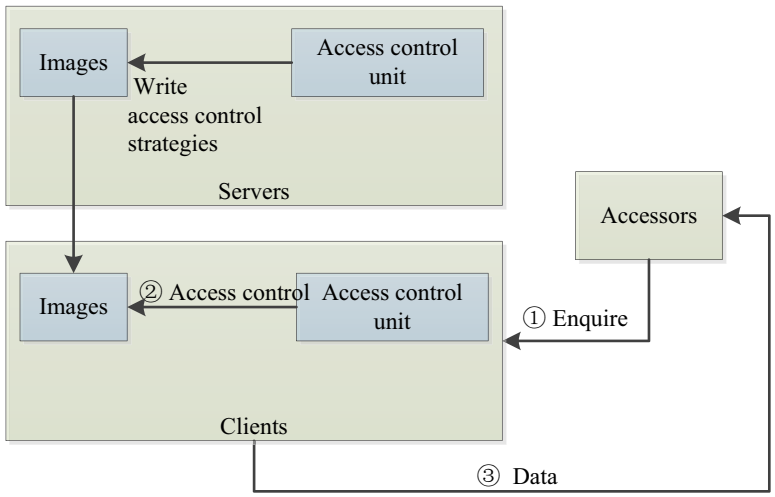

Figure 2. Existing traditional access control model.

Note that embedding methods for access control policies are independent with the above architecture. Watermarks or other associated tags can also be workable if they can reveal access control policies. In most cases, invisible watermarks may be preferred.

Access control policies are embedded with big data, and thus the access control unit is moved to clients for persistent control, regardless of how many times the data are re-distributed. Additionally, access control can be accomplished without assuming the availability of servers and networking connections, which also mitigates the workload of servers and shortens the access delay.

3.2. Attack Models

3.2.1. Transferring Attack

Existing access control models invite the transferring attack. In a transferring attack, if accessor "A" can access image "P", then accessor "A" can transfer image "P" to others, such as accessor "B". Thus, accessor "B" can easily gain the access privileges of accessor "A".

To tackle this attack, we propose the use of a watermark-based access control model where access policies are embedded with objects and move the access control unit from servers to clients.

Besides, transferring attacks cannot be accountable. That is, it is impossible to trace back to original leaking accessors if many accessors can access the same objects. In other words, the provenance of leakage is lost. To provide provenance, we can also rely on watermarks that can reveal the identification of originators or leakers.

Proposition 1. *For persistent access control, access control policies need to be associated with accessible objects, and the objects can only be accessed upon parsing policies at clients. Additionally, the objects need to return back to unaccessible status after the allotted time of authorized access.*

Proof. If objects do not retain unaccessible status after being accessed, others can also access those objects when they are transferred to others.

If access control policies are not associated with accessible objects, clients will not be able to enforce access policies. □

Proposition 2. *For the provenance of distributed data, data must carry the identification information of originators.*

Proof. If data do not carry any of the originators' identification information, the provenance of who distributes data cannot be determined. □

3.2.2. Distributed Denial of Service (DDoS) Attack

Traditional access control models rely on the availability of servers and access control units. The availability can be damaged by distributed denial of service (DDoS) attack. If servers or access control units cannot be accessed, access processes or services will be terminated. It is much easier to let clients be available than servers, thus access control that is migrated to clients will be more scalable and durable.

3.2.3. Coarse Access

In traditional access control models, servers are confronted with a large volume of data and access requests, and fine-grained access control will experience much difficulty due to workload. It is not fine-grained if access control is specific to an entire image, instead of for a specific area or layer in the image—especially for those images that have large size such as geographic graphs or remote sensing graphs. Traditional models may have to tackle fine-grained access by extra control, which further increases the overhead of servers.

3.2.4. Physical Copy Attack

In image big data distribution, the most difficult attack to defend against is physical copy attack, in which images are copied by physical manners such as screen capture or outside photo shooting. After accessors gain access to images, those images are totally displayed and out of (access) control. This attack must be tackled, especially if certain areas or layers in images must remain confidential. It cannot be defended against by access control because it is a kind of proactive defense before events.

This attack can be traced back by watermark-based schemes for further provenance, as that is a kind of reactive defense after events.

Proposition 3. *Physical copy attack cannot be defended against by any access control schemes, but it can be traced back to the source of image leakers, which is called provenance. The provenance can only be achieved by associated watermarks in images.*

Proof. As images can be uncovered and viewed by authorized accessors, physical copy attack such as screen capture and photo shooting is also possible.

The provenance can be achieved by embedding watermarks in images, as watermarks are also carried by images during and after physical copy attack.

Only when some watermarks associated with the identity of originators are embedded with uncovered images can the provenance of originators who exposed the images be accomplished from leaked images. □

3.3. Design Goals

We list design goals as follows: Design a novel access control flow that migrates the control unit from servers to clients. Design a watermark-based access control model that provides fine-grained access control for various areas or layers in a single image. Defend against attacks imposed by traditional access control models and propose a tailored design for big data sharing and redistribution of images with large sizes.

Remark 1. *Images can be downloaded only from servers who embed access policies into images via watermarks.*

Images can only be viewed via particular client tools, such as an image browser that can extract watermarks, parse watermark semantics into policies, and enforce access control policies before viewing. The context of watermarks can be recognized by corresponding clients.

Accessors may register their roles on servers at first, and their roles can be affirmed by client tools before viewing images.

The client tool can transparently decrypt images by asking for the correct keys. After accessors view their corresponding partial areas, those areas are encrypted again by client tools transparently.

If a hard copy of images is obtained by screen capture or photo shooting, watermarks in images can facilitate the trace back to the accessor who was the last authorized viewer.

4. Proposed Scheme

4.1. Basic Settings

We first describe a concrete process to explain our scheme, which consists of three steps as follows:

1. Accessors registration. Accessors register for data access on servers. They are assigned a role or multiple roles by servers.
2. Data publication. Servers who are data publishers or distributors embed access control policies via watermarks in data such as images. Data is published, in which certain areas or layers may be encrypted by secret keys related to control policies.
3. Client conformance. Accessors request images via particular client tools, such as image browsers. Client tools ask accessors to present their roles and secret keys. Client tools enforce control policies by parsing from watermarks that are embedded in images, and decrypt corresponding areas or layers in images by responding secret keys.

Obviously, data publication and client conformance are critical in the design. Next, we propose a hierarchical encryption model as a concrete scheme.

4.2. Hierarchical Key-Role-Area Access Control Model

The encryption (and decryption) of various areas in a single image can be conducted by the following proposed hierarchical models.

$HKRAGraph ::= \langle V, E \rangle$;
$V = \{KEY, ROLE, AREA\}$
$E = \{E_{k2k}, E_{r2k}, E_{a2r}\}$;
$E_{k2k} = \langle from, to \rangle, from, to \in KEY$;
$E_{r2k} = \langle from, to \rangle, from \in ROLE, to \in KEY$;
$E_{a2r} = \langle from, to \rangle, from \in AREA, to \in ROLE$.

1. Hierarchical Keys

   (a) $KEY ::= \langle l, c \rangle$, where $l \in \mathbb{N}$ is a key level, and $c \in \mathbb{N}$ is a key column. Keys should be classified into different levels. In other words, a key has two metrics: one is key level denoted as l, and the other is key column denoted as c.

   (b) $K2L : k \in KEY \rightarrow l \in \mathbb{N}$, where KEY is a set of keys; l is a natural number representing key level. It is a function. It does not need to be not one-to-one. That is, multiple keys may map to one level. It is on-to. We denote the $k \in KEY$ with level l as $k[l, \cdot]$. If multiple keys map to the same level l, we distinguish them as $k[l, c], c \in \mathbb{N}$.

   (c) $K2C : k \in KEY \rightarrow c \in \mathbb{N}$, where KEY is a set of keys; c is a natural number representing key column. It is a function. It does not need to be one-to-one. That is, multiple keys may map to one column index. It is on-to. We denote the $k \in KEY$ with index c as $k[\cdot, c]$. If multiple keys map to the same column c, we distinguish them as $k[l, c], l, c \in \mathbb{N}$.

   (d) $k[l+1, c] \Leftarrow g(k[l, c])$, where $k[l, c] \in KEY$ and $\forall l \in \mathbb{N}$. That is, $\forall l \in Set_l = \{\ell | \ell = K2L(k[l, c] \in KEY)\}$, $c \in Set_c = \{c' | c' = K2C(k[l, c])\}$. $g(\cdot)$ is a one-way function. It is computationally infeasible to obtain x from $g(x)$, where $x \in KEY$.

   (e) $k[j, c]$ can be computed from any $k[i, c]$ $(i < j)$ by $k[j, c] = g^{j-i}(k[i, c])$, where $\forall i, j \in Set_l, c \in Set_c$, $g^{m+1}(\cdot) = g(g^m(\cdot)), m \in Set_l, g^1(\cdot) = g(\cdot)$. Similarly, $\forall j > i$, $k[j, c]$ can be computed from $k[i, c]$ by $k[j, c] = g^{j-i}(k[i, c])$, where $\forall i, j \in Set_l, c \in Set_c$, $g^{m+1}(\cdot) = g(g^m(\cdot)), m \in Set_l, g^1(\cdot) = g(\cdot)$.

   Simply speaking, a key with a larger key level can be derived from any key with smaller key levels in the same key column. If accessors possess a key of a smaller level, they can derive all keys with larger key levels in the same key column. Thus, a larger-level key can decrypt the data encrypted by a smaller-level key, but not inversely.

2. Hierarchical Roles

   (a) $ROLE ::= \langle l, c, u \rangle$, where l is a key level, c is a key column, and u is an identification to distinguish multiple roles for the same key. As multiple roles may map to the same key with $k[l, c]$, multiple identifications (e.g., u) are required for the distinction of multiple roles.

   (b) $R2K : r \in ROLE \rightarrow k \in KEY$, where $ROLE$ is a set of roles; KEY is a set of keys. It is a function. It does not need to be one-to-one. That is, multiple roles may map to one key. We denote $r \in ROLE$ that maps to the same key $k[l, c]$ as $r[l, c, u], l, c, u \in \mathbb{N}$. $R2K(\cdot)$ is on-to. Simply speaking, multiple roles may be related to one key. Regarding the privileges for images, the mainly one is "read". A role with smaller (higher) levels can access all objects that can be accessed by roles with larger (lower) levels. Each role will be mapped to a key.

   (c) $R2L : r \in ROLE \rightarrow l \in \mathbb{N}$, where $ROLE$ is a set of roles; l is a natural number representing a key level. Note that $\forall r \in ROLE, R2L(r) \Leftarrow K2L(R2K(r))$. That is, roles are also hierarchically classified into different levels.

(d) $R2C : r \in ROLE \rightarrow c \in \mathbb{N}$, where $ROLE$ is a set of roles; c is a natural number representing a column number. Note that $\forall r \in ROLE, R2C(r) \Leftarrow K2C(R2K(r))$. This function returns a key index (in terms of key column) for a role, which can be used for guaranteeing derivative relationship between keys.

(e) $R2U : r \in ROLE \rightarrow u \in \mathbb{N}$, where $ROLE$ is a set of roles; u is a natural number representing users who are associated to the same key. Note that $\forall r_1, r_1 \in ROLE$, if $R2K(r_1) = R2K(r_2)$, then $R2U(r_1) \neq R2U(r_2)$.

The model proposed above is illustrated in Figure 3.

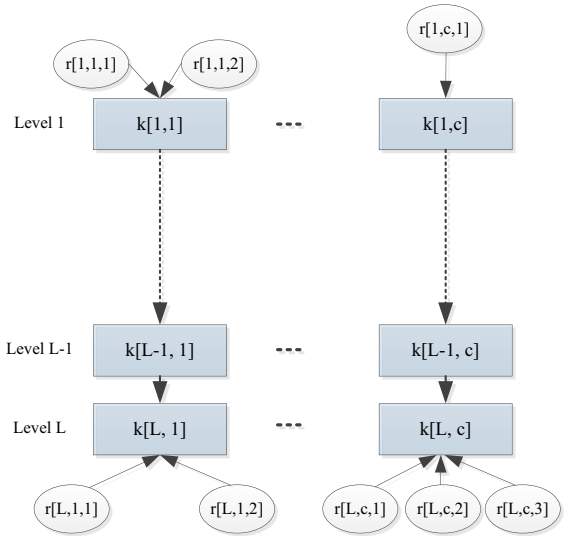

Figure 3. Hierarchical key and role model.

3. Differentiate Areas by Roles

(a) $AREA ::= \langle l, c, u, i \rangle$, where l is a key level; c is a column number; u is an identification to distinguish multiple roles for the same key; i is an identification to distinguish multiple areas for the same role. Note that $\cap_{l,c,u,i} a[l, c, u, i] = \emptyset$.

(b) $A2R : a \in AREA \rightarrow r \in ROLE$ is a function. It does not need to be one-to-one. That is, multiple areas may be assigned to one role. As r is a tuple with thre elements, a is a tuple with four elements.

(c) $A2K : a \in AREA \rightarrow k \in KEY$ is a function. It does not need to be one-to-one. Note that $\forall a \in AREA, A2K(a) \Leftarrow R2K(A2R(a))$.

(d) $A2L : a \in AREA \rightarrow l \in \mathbb{N}$. Note that $\forall a \in AREA, A2L(a) \Leftarrow R2L(A2R(a))$.

(e) $A2C : a \in AREA \rightarrow c \in \mathbb{N}$. Note that $\forall a \in AREA, A2C(a) \Leftarrow R2C(A2R(a))$.

(f) $A2U : a \in AREA \rightarrow u \in \mathbb{N}$. Note that $\forall a \in AREA, A2U(a) \Leftarrow R2U(A2R(a))$.

Remark 2. *Note that, $AREA$ can also be replaced by $LAYER$. In geographic images, there may be multiple layers in a single image.*

$a \in AREA$ could be any shapes (e.g., circles or rectangles), which are independent of the design of this paper. The details on areas can be embedded in watermarks, such as one-point locations with two rectangular edges. Areas for different roles can be overlapped. For different roles with the same R2K, the areas may be different and one area information for one role may not be available for the other role.

If we remove the constraints of R2K from a function to any mapping, then one role may map to multiple keys.

The proposed access control model is illustrated in Figure 4.

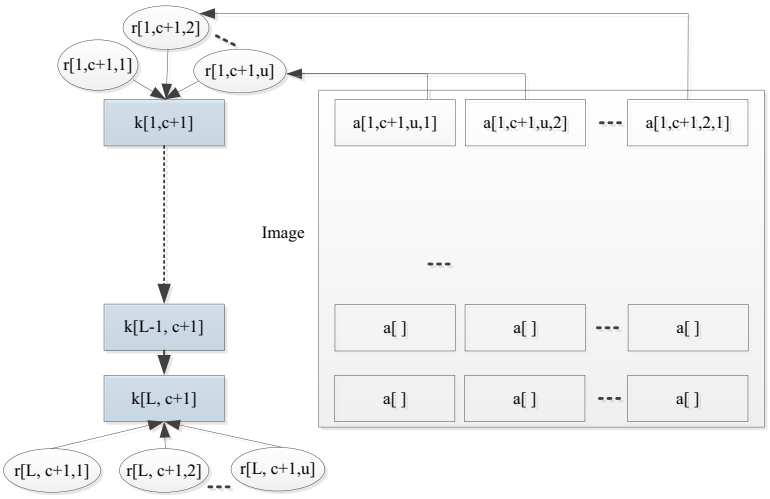

Figure 4. Hierarchical key-role-area access control model (*HKRAGraph*).

4.3. Image Publication

Images can be processed before publication as follows:

1. Servers select an image to publish. Corresponding areas (e.g., $a \in AREA$) in this image are split according to security concerns and assigned to different roles. Areas are layered into different security levels, such that roles who can access higher security level (with larger key level) will be able to access lower security levels (with smaller key level). Servers formulate access control strategies by $ACL ::= \langle ROLE, AREA \rangle$, where $\forall a \in AREA, \exists r = A2R(a) \in ROLE$.
2. Servers code access control strategies into watermarks and embed them into published images. For example, QR codes can be used as watermarks, and strategies are coded into QR codes.
3. Servers maintain a table for the image $TBL ::= \langle a \in AREA, f(A2K(a)) \rangle$, and encrypt specific areas in images with corresponding keys. For example, servers encrypt a by $f(A2K(a))$. $f(\cdot)$ is a one-way function. $f(A2K(a))$ instead of $A2K(a)$ is stored for better confidentiality. $A2K(\cdot)$ is initialized by servers in *HKRAGraph*.
4. $\forall a \in AREA$ in this image, a is encrypted by $f(A2K(a))$, and note that all $K2C(A2K(a))$ are identical.
5. $\forall a_1, a_2 \in AREA$ in an image, we have $A2C(a_1) = A2C(a_2)$. Simply speaking, for all areas in one image, encrypt keys must be in the same column index.
6. $\forall a_1, a_2 \in AREA$ in an image, if $A2L(a_1) = A2L(a_2)$, then $A2K(a_1) = A2K(a_2)$ due to $A2C(a_1) = A2C(a_2)$.

4.4. Client Conformance

Client conformance for access control can be processed as follows:

1. Accessors request images via a particular client tool (e.g., image browser).
2. The browser prompts to ask for and obtain a secret key k' and a role r' corresponding to an accessor.
3. The browser extracts a QR code, obtains access control strategies (i.e., $ACL ::= \langle ROLE, AREA \rangle$). All $a \in ACL.AREA$ are obtained for $r' \in ACL.ROLE$. That is, $A2R(a) = r'$.
4. The browser computes $f(k')$, and decrypts all areas for r' (i.e., a). Note that the key is not stored in the browser, and only $f(k')$ is computed temporarily by the browser and destroyed after browsing.
5. Calculate all $j > l$, $k[j,c] \Leftarrow g^{j-l}(k[l,c])$, $k[l,c] = k'$ and decrypt left areas at lower levels. That is, $a \in ACL.AREA, A2R(a) \neq r'$ by $k[j,c]$.
6. The browser displays all a to the accessor.
7. Accessors close the browser, and the browsed image returns to its original encryption status.

Remark 3. *Servers will maintain consistency with client tools for function $f(\cdot)$ (i.e., the same $f(\cdot)$). Once the consistency is retained, $f(\cdot)$ can be evolved further regularly to provide forward security. Alternatively, an extra pairwise key (e.g., key_p) between servers and client tools can be introduced into $f(\cdot)$ as $f(\cdot,\cdot)$ (e.g., $f(\cdot, key_p)$). We stress that client tools do not locally and permanently store accessor keys. Instead, decryption keys for encrypted areas in images are computed temporally upon browsing.*

4.5. Case Study

It is a trend to incorporate multiple maps from one location into one map as multiple layers. For a better explanation, we separate a combined map with multiple layers into three individual maps. In this case study, three maps of Shanghai are displayed in Figure 5 [23], which includes a remote sensing image, a geologic map, and a city planning map. These three maps describe three aspects of the same location. A combinative map can provide various aspects of one location in one map by multiple layers, which facilitates fast linkages to relevant information within one area.

Figure 5. A combinative map of Shanghai with multiple layers. The first one is sensing image. The second one is a geologic map. The third one is a city planning map.

The security levels of roles and corresponding layers are embedded into maps as watermarks, and thus access control strategies can be obtained from distributed maps without consulting servers. Accessors present their roles to a dedicated client tool such as an image browser, and specific areas that can be accessed by presented roles will be determined by the client tool.

In one map, accessible areas are encrypted by corresponding keys (e.g., a_i is encrypted by k_i ($i = 1, 2, ..., n$)). Only someone who presents the correct key (e.g., k_i) can view the corresponding encrypted areas (a_i). We also provide a kind of hierarchical access by hierarchical encryptions for areas. That is, keys at lower security levels can be derived by keys at higher security levels (e.g., k_{i+1} can be derived by k_i). Thus, areas for roles in lower security levels can also be decrypted and viewed by roles with higher security levels. Upon request for images by an accessor, the image browser will prompt the accessor to present their key (e.g., k_i). The image browser will compute $f(k_i)$ and use it to decrypt corresponding areas.

In combinative maps, one area consists of multiple aspects presented in layers. For example, geology, remote sensing, and city planning are three layers of a single city, Shanghai. Some accessors may only be able to access one layer among them. Accessors present their roles and keys to reveal corresponding layers.

5. Security and Performance Analysis

5.1. Security Analysis

Defending Against Transferring Attack. Images are encrypted by designated keys related to corresponding roles or accessor identifications, and accessors must present the correct keys to enable client tools to decrypt images for browsing. Encrypted images cannot be decrypted without keys, even if images are transferred to others again. Moreover, decrypted images can only be decrypted and displayed in client tools. Images will return to their original encrypted status after browsing.

The control unit migrates to client tools and it maintains control even though images are redistributed again. The control policies are associated with images as watermarks, which specify what areas can be viewed for given roles. The decryption can only occur upon browsing, and the encrypted area returns back to confidential status after images are browsed in the client tools. That is, the encrypted areas (layers) are transparently decrypted and ephemerally displayed upon browsing.

Defending Against DDoS Attacks. As access control logics are embedded in watermarks together with images, client tools can control access policies without consulting servers and relying on networking connections. Thus, DDoS attacks for servers and networking connections are not workable.

Defending Against Coarse Access. Our model can differentiate the access privileges for various areas in a single image, and similarly, further access control for various layers in a single area are also possible iteratively.

Defending Physical Copy Attack. As visible watermarks such as QR codes or invisible watermarks are incorporated with images, anyone who obtains physical copies of images by screen capture or outside camera shooting will be traced back by watermarks. The roles and identifications can be revealed by decrypted areas in captured images and control policies in watermarks.

Proposition 4. *It is hard to compute k_j from k_i if $k_j = g(k_i)$, where $f(\cdot)$ is a one-way function.*

Proof. Straightforward. We use a one-way function to drive keys in lower security levels from keys in higher security levels. As the function is one-way, the derivation of keys will be also one-way. That is, it is hard to compute x from $g(x)$. □

5.2. Performance Analysis

Computation Cost. The major computation in the scheme are as follows: encoding and decoding watermarks, encrypting and decrypting areas in images, and one-way function computation. However,

encoding watermarks can be conducted only one time. Encryption is conducted one time for each image, and decryption is conducted one time for each instance of image browsing. Note that encryption and decryption cannot be avoided for image access control, as some contents must be encrypted for confidentiality. One-way function computation is lightweight (e.g., cryptographically secure hash function).

Higher Access Throughput and Less Access Delay. The access control policies are embedded into watermarks and distributed with images, thus it is not mandatory to consult servers for corresponding areas that can be accessed. This improves the scalability of data access. Besides, the access delay is decreased due to the absence of consulting communications latency between servers and clients.

Efficiency. A balance between servers and clients is preferred, instead of only relying on servers. Servers only need to attach a watermark to an image and encrypt designated areas upon data publication, which can be accomplished in a batch. Client tools only need to decode a watermark and decrypt corresponding areas. The decryption is conducted at the client side, which is much more lightweight than at the server side. The encryption and decryption are mandatory because some areas are confidential.

Convenience. The deployment is convenient. Particular client tools can be deployed as middle-ware over normal image browsers. Besides, communication channels and networks are not required, which brings more convenience for accessors.

QR codes can be used for fast generation and decoding of watermarks. It presents the advantages of large capacity, fault tolerance, easy generation, and fast decoding. Thus, the overhead of attaching and decoding watermarks is manageable.

Table 1 compares the advantages and disadvantage between ours and existing schemes.

Table 1. The comparisons between Proposed Scheme and other existing related work.

Performance	Our Scheme	R. Wolfgang [7]	R. Kountchev [9]	RVM [10]	A. Phadikar [11]
1. Hierarchical access control	✓	✗	✓	✗	✗
2. Code access control strategies in watermarks	✓	✗	✗	✗	✗
3. Access control can take effect without servers	✓	✗	✗	✗	✗
4. Watermarks are used for access control	✓	✗	✓	✓	✓
5. Record modifiers in watermarks	✓	✗	✗	✗	✗
6. Copyrights protection	✓	✓	✓	✓	✓
7. Quality access control	✓	✗	✗	✓	✓
8. Fine-grained	✓	✗	✗	✗	✗

6. Conclusions

In this paper, we propose a watermark-based access control model. In contrast to current access control methods, we attach access control strategies within accessed objects (e.g., images) as watermarks, instead of storing access control strategies on servers. Our proposed model makes it possible to let accessors view images without accessing servers. This can ease the burden of servers and shorten the access delay. In addition, our model also defends against several dedicated attacks for accessing big image data, such as transferring attack, DDoS attack, coarse access, and physical copy attack. Moreover, we also propose a hierarchical key-role-area access control model. In this model, multiple areas in an image can be mapped to one role, and each role is associated with a hierarchical key. Hierarchical keys are classified into levels and keys at higher security levels can derive keys at

lower security levels. Thus, various areas that can be accessed by different roles in one image can be encrypted by hierarchical keys. Because of the above key-role-area model, fine-grained access control can be achieved in a more complicated and customized manner. Especially, the above method can also be applied for different layers in a single image (e.g., geographic maps). Furthermore, further traceability of image leakage (e.g., areas, layers) becomes possible due to embedded watermarks.

Author Contributions: Conceptualization, W.R.; Methodology, W.R. and Y.R.; Software, J.G.; Validation, T.Z.; Formal Analysis, J.G. and T.Z.; Investigation, J.G.; Resources, J.G.; Data Curation, J.G.; Writing—Original Draft Preparation, J.G.; Writing—Review & Editing, W.R., Y.R. and T.Z.; Visualization, J.G.; Supervision, W.R.; Project Administration, W.R.; Funding Acquisition, W.R. and T.Z.

Funding: The research was financially supported by Major Scientific and Technological Special Project of Guizhou Province under Grant No. 20183001, the Open Funding of Guizhou Provincial Key Laboratory of Public Big Data under Grant No. 2017BDKFJJ006, Open Funding of Hubei Provincial Key Laboratory of Intelligent Geo-Information Processing with under Grant No. KLIGIP2016A05, and National Natural Science Foundation of China under Grant No. 61502362.

Acknowledgments: We also thank the comments from Y.C., W.J., Z.K. and M.L.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Facebook Says Cambridge Analytica May Have Gained 37 m More Users' Data. Available online: https://www.theguardian.com/technology/2018/apr/04/facebook-cambridge-analytica-user-data-latest-more-than-thought (accessed on 20 June 2018).
2. Xiong, H.; Choo, K.K.R.; Vasilakos, A.V. Revocable Identity-Based Access Control for Big Data with Verifiable Outsourced Computing. *IEEE Trans. Big Data* **2017**, *99*, 1. [CrossRef]
3. Xiao M.; Wang, M.; Liu, X.; Sun, J. Efficient distributed access control for big data in clouds. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, 26 April–1 May 2015; pp. 202–207.
4. Wang, Y.; Doherty, J.F.; Van Dyck, R.E. A wavelet-based watermarking algorithm for ownership verification of digital images. *IEEE Trans. Image Process.* **2002**, *11*, 77–88. [CrossRef] [PubMed]
5. Moulin, P. The role of information theory in watermarking and its application to image watermarking. *Signal Process.* **2001**, *81*, 1121–1139. [CrossRef]
6. Gunjan, R.; Laxmi, V.; Gaur, M.S. Detection attack analysis using partial watermark in DCT domain. In Proceedings of the Fifth International Conference on Security of Information and Networks, New York, NY, USA, 25–27 October 2012; pp. 188–192.
7. Wolfgang, R.B.; Delp, E.J. A watermark for digital images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 19 September 1996.
8. Wong, P.W. A public key watermark for image verification and authentication. In Proceedings of the 1998 International Conference on Image Processing (ICIP98) (Cat. No.98CB36269), Chicago, IL, USA, 4–7 October 1998; pp. 455–459.
9. Kountchev, R.; Milanova, M.; Kountcheva, R. Content protection and hierarchical access control in image databases. In Proceedings of the 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), Madrid, Spain, 2–4 September 2015; pp. 1–6.
10. Yang, H.; Yin, J. A secure removable visible watermarking for BTC compressed images. *Multimed. Tools Appl.* **2015**, *76*, 1725–1739. [CrossRef]
11. Phadikar, A.; Maity, S.P.; Delpha, C. Data hiding for quality access control and error concealment in digital images. In Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11–15 July 2011; pp. 1–6.
12. Phadikar, A.; Maity, S.P. A Cost Effective Scheme for Content Verification and Access Control of Quality of an Image. In Proceedings of the 2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems, Kharagpur, India, 8–10 December 2008; pp. 1–6.
13. Datta, K.; Gupta, I.S. Partial encryption and watermarking scheme for audio files with controlled degradation of quality. *Multimed. Tools Appl.* **2013**, *64*, 649–669. [CrossRef]

14. Asikuzzaman, M.; Pickering, M.R. An Overview of Digital Video Watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *99*, 1. [CrossRef]
15. Van Gasselt, S.; Nass, A. Planetary Map Data Model for Geologic Mapping. *Cartogr. Geogr. Inf. Sci.* **2011**, *38*, 201–212. [CrossRef]
16. Han-fa, X.; Bing-liang, C.; Li-lin, X. An Mixed Access control method Based on Trust and Role. In Proceedings of the 2010 Second IITA International Conference on Geoscience and Remote Sensing, Qingdao, China, 28–31 August 2010; pp. 552–555.
17. Kim, J.; Jeong, D.; Baik, D.K. A Multi-layer based Access Control Model for GIS Mobile Web Services. In Proceedings of the 2009 Digest of Technical Papers International Conference on Consumer Electronics, Las Vegas, NV, USA, 10–14 January 2009; pp. 1–2.
18. Kirkpatrick, M.S.; Damiani, M.L.; Bertino, E. Prox-RBAC: A proximity-based spatially aware RBAC. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA, 1–4 November 2011; pp. 339–348.
19. Ma, F.; Gao, Y.; Yan, M.; Xu, F.; Liu, D. The fine-grained security access control of spatial data. In Proceedings of the 2010 18th International Conference on Geoinformatics, Beijing, China, 18–20 June 2010; pp. 1–4.
20. Zhang, A.; Ji, C.; Bao, Y.; Li, X. Conflict Analysis and Detection Based on Model Checking for Spatial Access Control Policy. *Tsinghua Sci. Technol.* **2017**, *22*, 478–488. [CrossRef]
21. Kao, Y.W.; Luo, G.H.; Lin, H.T.; Huang, Y.K.; Yuan, S.M. Physical Access Control Based on QR Code. In Proceedings of the 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, 10–12 October 2011; pp. 285–288.
22. Melgar, M.E. V.; Zaghetto, A.; Macchiavello, B.; Nascimento, A.C. CQR codes: Colored quick-response codes. In Proceedings of the 2012 IEEE Second International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 3–5 September 2012; pp. 321–325.
23. Available online: https://zhfw.tianditu.gov.cn/ (accessed on 20 June 2018).