第 36 卷 第 3 期
2007 年 8 月

测 绘 学 报
ACTA GEODAETICA et CARTOGRAPHICA SINICA

Vol. 36, No. 8
Aug., 2007

文章编号:1001-1595(2007)03-0309-07 中图分类号:P208 文献标识码:A

Security Mechanism for Distributed GIS Spatial Data Based on Object-based Storage

YU Zhan-wu[1], LI Zhong-min[1], ZHENG Sheng[2], LI De-ren[1]

(1. *State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan* 430079, *China*; 2. *School of Electronic Information, Wuhan University, Wuhan* 430079, *China*)

基于对象存储的分布式 GIS 数据安全机制

喻占武[1], 李忠民[1], 郑 胜[2], 李德仁[1]

(1. 武汉大学 测绘遥感信息工程国家重点实验室,湖北 武汉 430079; 2. 武汉大学 电子信息学院,湖北 武汉 430079)

摘 要:根据 OSD-2 标准,提出一种安全机制来保证基于对象存储的分布式 GIS 空间数据的安全。在本安全机制中,采用的安全模型是基于信任状的访问控制系统,主要由 OSD 设备、安全管理器、策略/存储管理器和 GIS 服务器组成。该安全机制主要包含 3 个协议:GIS 服务器-安全管理器协议、安全管理器-OSD 设备协议和 GIS 服务器-OSD 设备协议。命令的传输和数据的访问都要进行认证。这 3 个协议有效预防了各种网络攻击手段的攻击,保证了分布式 GIS 空间数据的安全共享。

关键词:地理信息系统;基于对象存储;安全机制;信任状

Abstract: Massive Spatial data is the very core of current GIS and generally is distributed at different places. A new generation storage model for distributed GIS spatial data based on Object-Based Storage (OBS) has been constructed, which gives an integrated solution to both offer high-performance storage services and secure data sharing. In this model, GIS server, metadata server and storage device are separated, so it is very important to construct a security mechanism for securing distributed GIS spatial data.

In this paper, a security mechanism for distributed GIS spatial data is presented based on OBS after analyzing Object-based Storage Device (OSD) security model described in OSD-2 standard. In this mechanism, the security model is a credential-based access control system composed of the following components: ① an OSD device (OBSD); ② a security manager; ③ a policy/storage manager; ④ a GIS server. Commands transfer and data access both need be authorized. The mechanism is composed of three security protocols: GIS Server-Security Manager protocol, Security Manager-OBSD protocol and GIS Server-OBSD protocol. These three protocols maintain integrity, confidentiality and privacy of distributed GIS spatial data.

Key words: GIS; object-based storage; security mechanism; credential

1 Introduction

Massive spatial data is the very core of GIS, and has taken on obvious characteristics of multi-scale, multidimensional, distributed sharing, and increasing quickly[1], so it is very important to secure distributed GIS spatial data.

The model of current GIS spatial data storage and management is mainly spatial database based on ORDB (Object-Relation Database)[2~5], which physical storage devices generally use the RAID or other large capacity storage devices directly linked to the interior data bus of GIS server (In this paper, this model is called as spatial database storage model). In the spatial database storage model, data access, control and management are all run by GIS server, so it has some

收稿日期: 2006-01-26; **修回日期**: 2007-04-03
基金项目: 国家重点基础研究发展计划(973 计划)项目(2004CB318206)
作者简介: 喻占武(1969-),武汉人,教授,博士生导师,主要从事多媒体通信和海量空间信息存储研究。
E-mail: yuzhanwu@lmars.whu.edu.cn

difficulties in offering high-performance storage services and securing data sharing. GIS applications running in the GIS server manage spatial database by the means of authentication and authorization to offer securing distributed spatial data sharing.

The model of GIS spatial data storage based on Object-Based Storage (OBS) gives an integrated solution to both offer high-performance storage services and secure data sharing. In this model, GIS server, metadata server and storage device are separated, and spatial data is organized as a spatial object saved in Object-based Storage Device (OSD), so we present a security mechanism for distributed GIS spatial data based on OBS to secure massive spatial data.

2 GIS Spatial Data Storage Model Based on OBS

The network storage technologies used in the spatial database storage model mainly have Direct Attached Storage (DAS), Network Attached Storage (NAS) and Storage Area Network (SAN). DAS is a network storage architecture that the block-based storage devices are directly linked to the I/O bus of a computer or a server via SCSI or ATA/IDE, which can offer storage space extension and high-performance transfer for a single server which both offers network storage services for clients outside and provides access control and security policy for whole DAS system. For NAS, the storage subsystem is attached to a network of servers and file requests are passed through a parallel file system to the centralized storage device[6,7]. NAS uses a simplified operating system, which is specified to use for storage, to realize file sharing, and can directly link to the switch of WAN or the hub of other LANs, and usually uses access control and security policy at file granularity[8]. SAN uses a dedicated network to provide an any-to-any connection between processors and storage devices[6], usually connected by Fiber Channel or iSCSI[9]. SAN is an unattached data storage network, and the data transfer rate is very high inside the storage network. But SAN can't realize the file sharing across platforms, and its operating system still lies in the server, the user can't directly access the storage network. Its access control and security policy are only implemented by storage devices and network switches, so the granularity is too big to realize the access control and storage security in an unsafe network.

Just like the spatial database storage model, the spatial object storage model also includes a user component and a storage component. In this model, different to the spatial database storage model, spatial data is organized as a spatial object saved in OSD but not in spatial database, and the user component and the storage component are separated. Fig. 1 is the comparison of spatial database storage model and spatial object storage model[10].

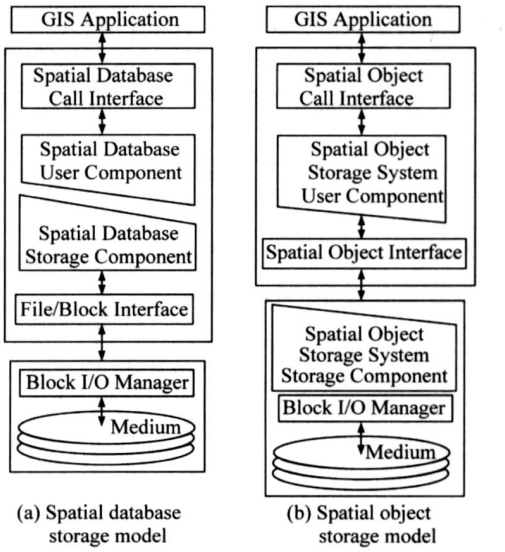

Fig. 1 Comparison of spatial database and spatial object storage model

The user component contains such functions as hierarchy management, naming and user access control, and may have the ability to influence the properties of object data through the specification of attributes mechanisms. GIS application communicates with the user component using spatial object call interface.

The storage component is offloaded to the storage device and the interface accessing the storage device changes from file/block interface to spatial object interface. It is focused on mapping spatial object to the physical organization of the storage media, and makes the decisions as to where to allocate storage capacity for individual data entities and managing free space. In addition to mapping data, the storage component maintains other information about the spatial objects that it stores (e.g., size, and usage quotas, and associated username) in attributes.

In order to separate access paths of control, management and data, GIS server, spatial metadata server and the OBSDs (OSD devices) are self-existent. The architecture of GIS based on OBS is showed in Fig. 2. The OBSDs are the storage components of the system to be shared. Spatial object is stored in abstract containers by the OSD logical unit. Spatial object in the abstract containers is not addressable using LBAs (Logical Block Addresses). The OSD logical unit allocates space for spatial object and delivers a unique identifier to the GIS server. The GIS server uses the same unique identifier for subsequent accesses to the spatial object. Metadata server manages the metadata of spatial data and OSD, and the GIS server directly accesses an OBSD. In this way, 90 % of metadata management is distributed in the OBS-

Ds, so it avoids the bottleneck problem of metadata in traditional storage system[11~13].

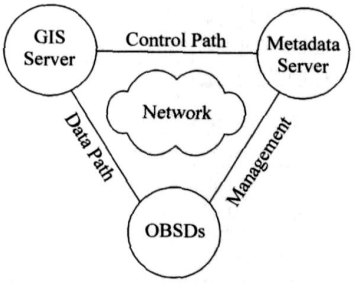

Fig. 2 GIS based on OBS architecture

In distributed GIS spatial data based on spatial database storage model, all of spatial data and metadata are accessed via metadata server, so the security of spatial data can be concentratively controlled in the metadata server, and it is realized easily by authentication and authorization. In distributed GIS spatial data based on OBS, GIS server, spatial metadata server and the OBSDs are separated and GIS server directly accesses the OBSDs, so it leads to some new security problems:

1. How to authenticate the GIS server for metadata server and OBSD?

2. How to secure the data exchanged between GIS server and OBSD?

3. How to protect spatial data against network attacks?

3 Proposed Solution

In order to solve the problems above, a security mechanism is discussed in details in this section.

3.1 The Security Model

According to the OSD security model described in OSD-2[11], a security model is given for distributed GIS spatial data based OBS, which is a credential-based access control system composed of the following components: an OBSD, a Policy/ Storage Manager, a Security Manager and a GIS Server. Fig. 3 shows the flow of transactions between the components of this security model.

In this security model, the OBSD and the Security Manager are trusted components[11,14,15]. The Security Manager generates credentials, including capabilities prepared by the Policy/ Storage Manager, for authorized GIS server. The Security Manager returns a Capability Key with each Credential. The Credential gives the GIS Server access to specific OSD components. The Capability Key allows the GIS Server and the OBSD to authenticate the commands and data they exchange with an Integrity Check Value.

The GIS Server requests credentials and capability keys from the Security Manager for the command functions it needs to perform and sends those capabilities in those credentials to the OBSD as part of commands that include an Integrity Check Value used as the Capability Key.

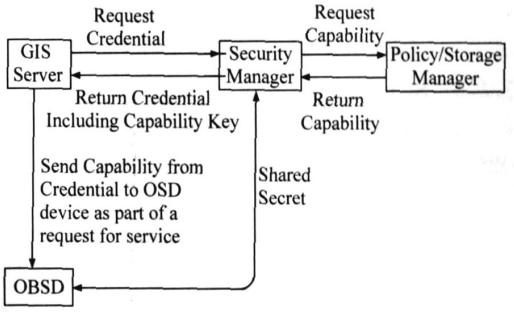

Fig. 3 Security model transactions

The Security Manager may authenticate the GIS Server, but the OBSD does not authenticate the GIS Server. It is sufficient for the OBSD to verify the capabilities and integrity check values sent by the GIS Server.

3.2 Parameters Definitions

3.2.1 Credential

The Credential is a data structure that is prepared by the Security Manager and protected by an Integrity Check Value that is sent to a GIS Server in order to grant defined access to an OSD logical unit for specific command functions performed on specific OSD objects. The Credential includes a Capability (Cap), an OSD System ID (ID_{OSD}) and a Credential Integrity Check Value (ICV_{Cre}), formulated as

$$C = [\, Cap + ID_{OSD} + ICV_{Cre} \,] \qquad (1)$$

1. The Capability is prepared by the Policy/ Storage Manager that the GIS Server copies to each CDB that requests the specified command functions. It is included in a CDB to enable the OBSD to verify that the sender is allowed to perform the command functions described by the CDB. The Capability Expiration Time, the security methods and algorithms are also specified in the Capability. In this security mechanism, the security method is the CMDRSP security method[7], and the security algorithm is HMAC-SHA1[16].

2. The Integrity Check Value (ICV) is a value computed using a security algorithm A (e.g., HMAC-SHA1 [5]), a secret key K, and an array of bytes B as

$$ICV \leftarrow (A, K, B) \qquad (2)$$

3. The ICV_{Cre} is used to protect the Credential against various network attacks. It is calculated by the Security Manager using the algorithm A specified in the Capability, the contents of the Credential except the ICV_{Cre} and secret key K_S shared by the Security Manager and the OBSD in the

CMDRSP security method as

$$ICV_{Cre} \leftarrow (A, K_S, Cap + ID_{OSD}) \quad (3)$$

In the security mechanism, ICV_{Cre} is used as the secret key K_R to calculate Request Integrity Check Value (ICV_{Req}) of GIS Server, and is also called as the Capability Key used to secure the communications between the GIS Server and the OBSD.

3.2.2 Request

The service Request (*Req*) sent by the GIS Server to the OBSD is placed in the Command Descriptor Block (CDB)[11], in which there are a Timestamp (T_s), a Capability, OSD system *ID* and a Request Integrity Check Value (ICV_{Req}) related to the OSD security.

1. The Capability is the contents of the Credential sent by the Security Manager to the GIS Server.

2. The Request Integrity Check Value ICV_{Req} is calculated by the GIS Server using the algorithm *A* specified in the Capability, the contents of the Request except the ICV_{Req} and secret key K_R in the CMDRSP security method. The ICV_{Req} is computed as

$$ICV_{Req} \leftarrow (A, K_R, Req - ICV_{Req}) \quad (4)$$

3.2.3 Response

The Response (*Res*) sent by the OBSD to the GIS Server to response the Request sent by the GIS Server to the OBSD, including a Response Integrity Check Value ICV_{Res} related to the OSD security.

The Response Integrity Check Value (ICV_{Res}) is calculated by the OBSD using the algorithm *A* specified in the Capability, the contents of the Response except the ICV_{Res} and secret key K_R in the CMDRSP security method. The ICV_{Res} is computed as

$$ICV_{Res} \leftarrow (A, K_R, Res - ICV_{Res}) \quad (5)$$

3.3 Protocols Details

In our security mechanism, there are mainly three protocols: ① GIS Server-Security Manager protocol; ② Security Manager-OBSD protocol; ③ GIS Server-OBSD protocol. Those protocols are showed in Fig. 4.

3.3.1 The GIS Server-Security Manager Protocol

The protocol between the GIS Server and the Security Manager, shown in Fig. 4 ①, is rather straightforward:

1. The GIS Server asks for a Capability;

2. The Security Manager verifies the GIS Server permissions, generates a Credential including a Capability Key, and sends it to the GIS Server.

The Credential contains the key $K_R = ICV_{Cre}$, hence it must be sent to the GIS Server over a secure channel. To establish this channel and let the Security Manager identify the GIS Server, both of them should authenticate each other in a Kerberos or PKI system[17], which is not part of the protocol.

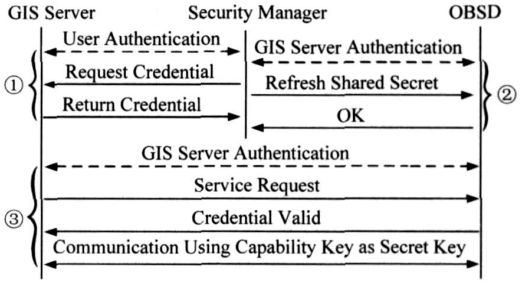

Fig. 4 Security mechanism including three protocols

3.3.2 The Security Manager-OBSD Protocol

The protocol between the GIS Server and the Security Manager is shown in Fig. 4 ②. It is the exchange protocol of the shared secret key K_S over a secure channel.

1. The Security Manager sends a new K_S to the OBSD along with their version number;

2. The OBSD stores the new K_S, and sends back an acknowledgement signal to the Security Manager.

3.3.3 The GIS Server-OBSD Protocol

The protocol between the GIS Server and the OBSD is shown in Fig. 4 ③. While received a Request sent by the GIS Server along with the capabilities in the Credential, the OBSD will

1. Verify the permission of the Request operations;

2. Check Capability Expiration Time in the Capability;

3. Reconstruct the Credential C' according to the Request and compute a new Credential Integrity Check Value ICV_{Cre}';

4. Compute a new Request Integrity Check Value ICV_{Req}' using the ICV_{Cre}' as the secret key;

5. Verify whether the ICV_{Req}' matches the ICV_{Req} of the Request.

If any of the checks mentioned above fails, the Request is denied. If all of them passes, $K_R' = K_R$, then the data exchanged between the GIS Server and the OBSD is encrypted by K_R.

4 Implementation Details

4.1 Credential Generation

The Security Manager has two roles: ① authenticating GIS Servers, authorizing their requests according to the system protection policy, and generating suitable credentials; ② refreshing the shared secret key K_S periodically[11]. The system protection policy and user authentication are outside of our security mechanism, only the Credential generation facility is discussed. The Security Manager and the Policy/ Storage

Manager are located in the Metadata Server.

The Security Manager keeps a table that contains the current encryption key, authentication key and key version of each of the OBSDs in the system. Given an OSD system *ID* and the permissible operation, the Security Manager generates the Credential as follows:

1. Forward the access requests from the GIS Server to the Policy/ Storage Manager. If the Policy/ Storage Manager denies the forwarded request an error shall be returned to the requesting GIS Server;

2. Insert the Capability returned by the Policy/ Storage Manager in the Credential;

3. Set the Credential OSD SYSTEM *ID* field to the value in the OSD system *ID* attribute in the Root Information attributes page of the OSD logical unit to which the Credential applies;

4. Create a random string K_S of length 128 bits;

5. Set the Capability KEY VERSION field to the number of the secret key K_S used to compute the Credential Integrity Check Value;

6. Set the Capability INTEGRITY CHECK VALUE ALGORITHM field to the value that specifies the algorithm *A* used to compute all integrity check values related to this Credential. The algorithm *A* selected in our security mechanism is HMAC-SHA1;

7. As specified by the maintained security policy information, modify other capability fields, including setting the CAPABILITY EXPIRATION TIME field to a value that is consistent with the policy;

8. Compute the Credential Integrity Check Value as described in 3.2.1, placing the result in the CREDENTIAL INTEGRITY CHECK VALUE field in the Credential;

9. Concatenate the Capability, the OSD system *ID* and the ICV_{Cre} of length 96 bits, thus generating the Credential.

10. Return the Credential thus constructed to the GIS Server with the Credential Integrity Check Value serving as the Capability Key.

4.2 Credential Verification

4.2.1 Reconstructing Credential

Before verifying the Credential associated with a CDB, the OBSD reconstructs a Credential C' from the Capability in the CDB by:

1. Copying the value in the OSD system *ID* attribute in the Root Information attributes page to the OSD SYSTEM *ID* field of the reconstructed Credential;

2. Copying the Capability from the CDB to the reconstructed Credential.

The CREDENTIAL INTEGRITY CHECK VALUE field is not used in a reconstructed Credential.

4.2.2 Verifying Credential

The OBSD verifies the Credential associated with a CDB by:

1. Reconstructing the Credential containing the Capability as described in 4.2.1;

2. Computing the Credential Integrity Check Value (ICV_{Cre}') for the reconstructed Credential C' using the algorithm *A* specified by the INTEGRITY CHECK VALUE ALGORITHM field in the capability, the contents of the reconstructed Credential C', and the secret key K_S shared with the Security Manager, formulated as

$$ICV_{Cre}' \leftarrow (A, K_S, C') \tag{6}$$

3. Recomputing a new Request Integrity Check Value (ICV_{Req}') using:

(1) The algorithm *A* specified by the INTEGRITY CHECK VALUE ALGORITHM field in the Capability;

(2) Based on the Request contents of the CDB and the REQUEST INTEGRITY CHECK VALUE ALGORITHM field set to zero;

(3) The Credential Integrity Check value ICV_{Cre}' computed in step 2 as the secret key K_R';

And formulated as

$$ICV_{Req}' \leftarrow (A, ICV_{Cre}', R) \tag{7}$$

4. Verifying that the Request Integrity Check Value ICV_{Req}' matches the contents of the REQUEST INTEGRITY CHECK VALUE field in the CDB. If the contents in the REQUEST INTEGRITY CHECK VALUE field in the CDB do not match the computed ICV_{Req}', the command shall be terminated with an Illegal Request status.

When the validation of a Credential is successful, there are $ICV_{Req}' = ICV_{Req}$ and $ICV_{Cre}' = ICV_{Cre}$.

4.2.3 Invalidating Credential

The Security Manager may invalidate the credentials for one OSD object by requesting that the Policy/ Storage Manager change the policy access tag attribute in the policy/ security attributes page associated with that OSD object to a value other than the policy access tag value that is contained in the credential's Capability.

The Security Manager may invalidate credentials for an entire partition by updating the working key version used to compute the credential integrity check value in those credentials.

4.3 Encrypting Communication

When the OBSD successfully verifies the Credential, ICV_{Cre}' equals to ICV_{Cre}, so there is an equation as

$$K_R = K_R' \tag{8}$$

K_R and K_R' are used as the single secret key to secure

the communications between the GIS Server and the OBSD. The GIS Server encrypts the Request using K_R is described above. The OBSD uses K_R' to secure the Data returned the GIS Server as follows:

1. The OBSD computes a Response Integrity Check Value (ICV_{Res}) using the algorithm A specified by the INTEGRITY CHECK VALUE ALGORITHM field in the Capability, the Response (Res) except the ICV_{Res} and the single secret key K_R', formulated as

$$ICV_{Res} \leftarrow (A, K_R', Res - ICV_{Res}) \quad (9)$$

2. The OBSD adds the ICV_{OSD} computed in step 1 into the Response and sent the Response including the ICV_{OSD} to the GIS Server;

3. When received the Response sent by the OBSD, the GIS Server recomputes an new Response Integrity Check Value (ICV_{Res}') using the algorithm A specified by the INTEGRITY CHECK VALUE ALGORITHM field in the Capability, the Response except the ICV_{Res} and the single secret key K_R, formulated as

$$ICV_{Res}' \leftarrow (A, K_R, Res - ICV_{Res}) \quad (10)$$

4. Verifying that the ICV_{Res}' matches the contents of the RESPONSE INTEGRITY CHECK VALUE field in the Response sent by the OBSD.

5 Security Analysis

5.1 Preventing Attacks

Security valuation of a security mechanism generally includes three aspects: authenticity, confidentiality and integrity. In order to guarantee the authenticity, confidentiality and integrity of the GIS spatial data stored in the OBS, the security mechanism must prevent various attacks such as eavesdropping, tampering, replaying, delaying, impersonating or masquerading, credential reusing. Our security mechanism can prevent those attacks by those means analyzed as follows:

1. Authenticity—it makes possible that the message receiver is capable of verifying the identity of the message sender, hence preventing that likely intruders inject malicious data into the OBS. In the security mechanism, the Security Manager and the GIS Server should authenticate each other in a PKI system. In order to guarantee authenticity, the mechanism frequently used is the Credential Integrity Check Value ICV_{Cre} calculation. ICV_{Cre} is a function that uses a private key and the proper message to obtain a distinguished a Capability, which may both be used in the authentication and the verification of the message integrity.

Impersonating or Masquerading attack forms a serious security risk in the OBS. For example, a malicious guest to the OBS may impersonate himself as the GIS Server, and then he may be able to access or destroy the GIS spatial data. Impersonating threats are mitigated by applying strong mutual authentications between the Security Manager and the GIS Server.

2. Confidentiality—it ensures that the content of the message is accessed only by authorized party.

Cryptography algorithms generally are used for guarantee data confidentiality. In our security mechanism, we uses K_R and K_R' as the single secret key to secure the communications between the GIS Server and the OBSD.

Eavesdropping is the most common and simple attack to a network storage system. Confidentiality is a security service against eavesdropping. The GIS spatial data must be encrypted using the single secret key to secure the communication between the GIS Server and the OBSD so that others who don't know the key will not disclose the content of the communication.

3. Integrity—it guarantees that should a message have its content modified during the transmission, the receiver is able to identify these alterations. Integrity is security requirement against tampering, replaying, delaying and reusing.

Tampering is the attacker actively modifies the Credential to fool the other party and obtain some benefit. It assures that the traffic is not damaged or modified by other unauthorized party. The Credential Integrity Check Value ICV_{Cre} protects the integrity of the GIS Server's Request. Tamper or forger can't calculate the right ICV_{Cre} without the secret key K_S, then can't tamper or forge the Capability, the Request or the Timestamp. ICV_{Cre} also protects the integrity of the Credential. If the GIS Server tampers or forges the Credential, the K_R' calculated by the OBSD using the secret key shared with the Security Manager doesn't equal to K_R, so the new Request Integrity Check Value ICV_{Req}' calculated by the OBSD using K_R' doesn't equals to the Request Integrity Check Value ICV_{Req} in the CDB sent by the GIS Server. The Request Integrity Check Value ICV_{Req} is the unique identifier of the GIS Server's Request. The OBSD keeps a table of received ICV_{Req} to verify whether the Request is implemented. If the Request is implemented, the OBSD refuses to offer interrelated service.

Replaying attack is the attack that attacker replays some previous eavesdropped packets to other parties. To prevent these attacks, we typically use nonce or timestamp to guarantee the freshness of the Credential.

The Timestamp of the Request is used to rebel against the delaying attack. The OBSD builds up a time window. Once the Timestamp of the CDB is out of the window, the OBSD refuses to offer interrelated service.

Credential reusing is permitted in order to avoid the GIS Server to apply the same Credential time after time. The val-

idation time of the Credential is insured by using the Capability Expiration Time in the Capability.

In a summary, the analysis shows the security mechanism we proposed for distributed GIS spatial data based on object-based storage is efficient and secure.

5.2 Credential Processing Speed

While a large amount of users parallel accessing the GIS spatial data storage system based on OBS, it is vital that the cost of generating and verifying a credential suits the qualifications of the high performances of GIS spatial data storage. In order to estimate the performance of our security mechanism we ran some initial tests to measure the cost of generating and verifying a credential. Our prototype implemented on a Pentium-Ⅳ 2.4 GHz with 512 MB memory running Linux RedHat with kernel version 2.4.21. The tests results are as follows:

1. Generating credentials: 18 500 per second.
2. Verifying credentials: 17 600 per second.

The results show that the cost of generating and verifying a credential is very low, and it absolutely suits the qualifications of the high performances of GIS spatial data storage.

6 Conclusions

The use of object-based storage instead of traditional block based storage is an apparent tendency. We have described our design and implementation details of a security mechanism for distributed GIS spatial data based on object-based storage. Our security mechanism can be applied to both the distributed GIS spatial data storage and other massive information storage.

The authors plan to further explore the security mechanism for distributed GIS spatial data based on object-based storage to understand how object-based access control lists can be efficiently designed and implemented.

References:

[1] MA Rong-hua, HUANG Xing-yuan. Distributed Organization and Management of the Large Volume of Data in Large GIS [J]. Journal of Nanjing University (Natural Science), 2003, 39(6): 836-843. (马荣华,黄杏元. 大型 GIS 海量数据分布式组织与管理[J]. 南京大学学报(自然科学版),2003, 39(6): 836-843.)

[2] LI De-ren, LI Qin-quan. Study of a Hybrid Data Structure in 3D GIS [J]. Acta Geodaetica et Cartographica Sinica, 1997, 26(2):128-133. (李德仁,李清泉. 一种 3 维 GIS 混合数据结构研究[J]. 测绘学报,1997,26(2):128-133.)

[3] LI Qin-quan, LI De-ren. Research on the Conceptual Frame of the Integration of 3D Spatial Data Model [J]. Acta Geodaetica et Cartographica Sinica, 1998, 27 (4): 325-330. (李清泉,李德仁. 3 维空间数据模型集成的概念框架研究[J]. 测绘学报,1998, 27 (4): 325-330.)

[4] WANG Lei, ZHOU Yun-xuan. A Study of Three-dimensional GIS Object-Oriented Data Model and Its Application [J]. Acta Geodaetica et Cartographica Sinica, 2002, 31(3): 274-277. (王 磊,周云轩. 3 维 GIS 面向对象数据模型的研究与应用[J]. 测绘学报,2002, 31(3): 274-277.)

[5] DONG Kai, FANG Yu. Concept of Spatial Database Model and Its Architecture Research [J]. Geomatics World, 2004, 2 (2): 8-16. (东 凯,方 裕. 空间数据库模型概念与结构研究[J]. 地理信息世界,2004, 2(2): 8-16.)

[6] WANG Yi-jian, KAELI D. Execution-driven Simulation of Network Storage Systems [A]. 12th Proceedings of IEEE Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems [C]. Volendam: [s.n.],2004. 604-611.

[7] GIBSON G A, METER R V. Network Attached Storage Architecture [J]. Communications of the ACM, 2000, 43(11): 37-45.

[8] ZHU Ying-wu, HU Yi-ming. SNARE: A Strong Security Scheme for Network-attached Storage [A]. Proceedings of 22nd International Symposium on Reliable Distributed Systems [C]. Florence: [s.n.],2003. 250-259.

[9] MENON J, PEASE D A, REES R, DUYANOVICH L, HILLSBERG B. IBM Storage Tank—A Heterogeneous Scalable SAN File System [J]. IBM Systems Journal, 2003, 42 (2): 250-267.

[10] YU Zhan-wu, ZHENG Sheng, ZHANG Yi. New Generation Storage Model for GIS Spatial Data Based on Object-based Storage [A]. Proceedings of SPIE of The Fourth International Symposium on Multispectral Image Processing and Pattern Recognition[C]. Bellingham: [s.n.],2005.

[11] American National Standards Institute. SCSI Object-based Storage Device Commands-2 (OSD-2) [EB/OL]. http://www.t10.org/ftp/t10/drafts/osd/osd-r10.pdf, 2004-07-31.

[12] SAKAR K. An Analysis of Object Storage Architecture [J]. IEEE Computer, 2003, 2(3): 23-36.

[13] KHER V, KIM Y. Decentralized Authentication Mechanisms for Object-based Storage Devices [A]. Proceedings of Second IEEE International Security in Storage Workshop (SISW-2003) [C]. Washington: [s.n.], 2003. 1-10.

[14] AZAGURY A, CANETTI R, FACTOR M. A Two Layered Approach for Securing an Object Store Network [A]. First IEEE International Security in Storage Workshop [C]. Maryland: [s.n.]. 10-23.

[15] BELLARE M, CANETTI R, KRAWCZYK H. Message Authentication Using Hash Functions: The HMAC Construction [J]. RSA Laboratories' CryptoBytes, 1996, 2(1).

[16] PERLMAN R. An Overview of PKI Trust Models [J]. IEEE Network, 1999, 13(6): 38-43.

(责任编辑:丛树平)