International Journal of Security and Its Applications
Vol. 10, No. 4 (2016), pp.167-178
http://dx.doi.org/10.14257/ijsia.2016.10.4.17

An Integrity Protection Model based on Trusted Recovery Technology

Xiaojie Xu [1,2] and Lisheng Wang[1]

[1]*School of Electronics and Information,Tongji University,Shanghai,China*
[2]*Network Center,Shanghai Normal University Tianhua College,Shanghai,China*

Abstract

This paper firstly through IBAC, integration of TE and RBAC, the use of compensatory well-formed transaction is proposed, the integrity of the structure can be recovered partial malicious transaction monitoring machine model. In the partial revocation of constitutive affairs, for the operation of data and tracking the affected, with two recovery policies. Conservative recovery policy to stop system the recovery of normal transaction execution, by analyzing log file dependencies list, according to operation performed after first order, cancel each affected operation. Another optimistic recovery policy can be in the normal operation of the system at the same time, the establishment of compensation operation corresponding to the operation to recover, and submitted to the monitoring machine scheduling integrity. This method can recover the system to a secure state in the face of failures and improves the availability of the system. It provides an important exploration for the design and implementation of the trusted recovery mechanisms of high-level secure operating system.

Keywords*: Computer Security, Integrity Policy, Trusted Recovery, Access Control*

1. Introduction

Access control model has confidentiality, integrity and neutrality of the three categories. Confidentiality policy model is represented by BLP model[1]. Integrity policy model has BIBA model[2],CW(clark-wilson) model,TE(type enforcement) model[3] and DTE (domain type enforcement) model[4] *etc*.RBAC(role-based access control)model[5] as a representative of the neutrality model has no clear security policy goals, achieving the security required to ensure the confidentiality and integrity of the system through the configuration and construction method. With the development of the computer system and application environment changes, the original system supported by a security policy requirements to support a variety of policy model so far. Therefore, Multi-policy Integration and support framework has become the hotspot of current research. Such as SELinux[6] system using FLASK (flux advanced securitykernel) framework, is a security model through integration of the IBAC, RBAC and TE models.

In reference monitor model[7], subject to the object of access request is submitted to the reference monitor machine, by the monitoring machine according to security policy decision whether to allow the request. If it does not meet the requirements of the security policy, it will be directly rejected. Reference monitor is used to determine the user (program) reference to the system resources (data, program, and equipment), and the program running control is responsible for controlling the resources of user program. If found inconsistent data, it should be able to return to a consistent state prior to such modification. Therefore, Povey propose a model of support for transaction recovery, with introducing the concept of PTP(partially transformation procedures) and CTP(compensating well-formed transformation procedures) in CW model[8-9]. We call it the Povey 's Clark - Wilson model (PCW) . However, when PTP is a malicious transaction, how to eliminate the impact and the revocation of the corresponding

ISSN: 1738-9976 IJSIA

operation of the problem, the PCW model has only given a implementation rules, no specific recovery algorithm, it is difficult to operate.

In this paper, we combine the theory of reference monitor and trusted recovery[10], by constructing TE model,integrating IBAC, RBAC, TE and PCW,then proposes a formal model which can restore the integrity of Malicious Partially formed transaction by CTP, and gives the static and runtime recovery algorithm of malicious transactions. The model can achieve access control, undo malicious transactions, eliminating all data systems affected by malicious transactions, to re-establish a consistent system state, achieve the requirement of the trusted recovery in high-level secure operating system.

2 Preliminaries

2.1 CW Model

Clark-Wilson (CW) the integrity model is a milestone, it is a complete sense of the origin of the integrity of the goal, policy and mechanism. CW model is based on two concepts: well-formed transaction and separation of responsibility。The former to ensure data internal consistency, the user can only manipulate the data in the form of a constrained, and cannot be arbitrarily manipulate data. The latter refers to all the operations are divided into several parts, each part performed by different people. CW model has four elements:CDI(constrained data item),UDI(unconstrained data item),IVP(integrity verification procedure),TP(transformation procedure).

CDI: we must use the integrity model to protect the data items in the system.

UDI: Without integrity policy model control data items,system introduced new data as UDI, but UDI can then be converted into CDI.

IVP: Used to verify all CDI specification meet the integrity in the system (in IVP execution) process.

TP: CDI set can change when a valid state to another.

CW model has five Certification and four Enforcement total of nine rules, constitute the system integrity strategy model. CW model effectively expressed the integrity of three goals: to prevent unauthorized users to modify, to prevent improper changes to the authorized users,to maintain the data consistency

2.2 PCW Model

Povey think that user behavior can be divided into three categories: legal, questionable and dangerous. Legal behavior is explicitly allowed by the access control policy. Dangerous behavior refers to the access control policy explicitly prohibits such as malicious behavior. Questionable behavior may be legal, usually need to give the user the permissions are generally not given, but questionable behavior may also be malicious behavior.

For the questionable behavior, PCW model adopted an optimistic security policy, that most of the user's access requests are legitimate, and are not explicitly prohibited. Through review of user behavior, if the administrator once found have abnormal behavior, can use compensation operation to repair damage. In order to reach the goal of recovery, all of the user's behavior must be able to rollback or compensation.

PCW model propose the concept of partially-formed transaction, which does not ensure date integrity, compared to well-formed transaction.That is, the actual behavior of its own is not constrainted, but the system can return to the previous valid state by compensating transactions.

PCW model adds three types of elements on the basis of CW model:PTP,CTP,PCDI. PCW model is given for 7 enforcement rules and 4 certification rules. The specific rules are shown in Table 1.

Table 1 Enforcement and Certification Rules in PCW Model

Rule	Content
C1	IVPs must be certified to ensure that all data items are in a valid state at the time the IVP is run.
C2	All PTPs must be certified to provide a compensating TP that will return any modified CDI to a valid state.
E1	The system must ensure that only PTPs that have been certified against requirement C2 are allowed to run.
E2	The system must ensure that users can only use those PTPs for which they have been authorized.
E3	The system must authenticate the identity of each user.
E4	EachPTP must write to an append-only log all the information required to reconstruct and reverse the operation.
E5	Only an administrator is permitted to authorize users to access PTPs.
E6	CDIs which are acted on by PTPs must be marked as PCDIs.
C3	Compensating TPs must be certified to result in valid CDIs.
C4	The administrator must certify PCDIs as being valid CDIs or if invalid,must apply the compensating transaction to restore the PCDIs to valid CDIs.
E7	If a PTP on a PCDI is reversed via a compensating TP,then all subsequent PTPs to PCDIs that depend on this item must also be reversed

2.3 Recoverable Integrity Monitoring Model

The model reference monitors the access request of subject to object,and deny or grant the request in accordance with the security policy[11]. During access process, audit system is used to record access to log file, and the recovery information used to recover the operation is added in the form of additional written in stable storage. Assuming that the contents of the stable storage cannot be tampered with, and suddenly in such cases data is not lost when power supply drop. Recoverable integrity monitoring model is shown in Figure 1.

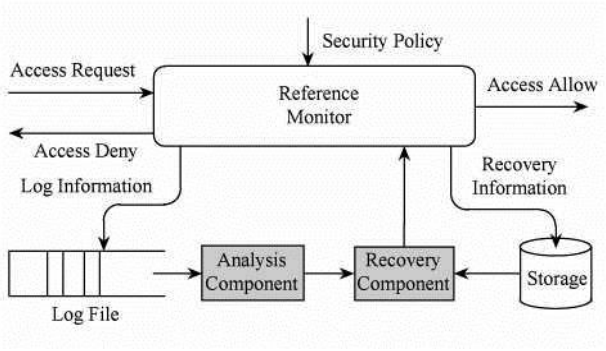

Figure 1. Model of Integrity Recovery Monitor

The monitor allows PTP to run, and when administrator finds PTP's operating exception, it can be recovered by CTP. According to requirements of E7 rules, in addition to undo PTP operation, also should be to rely on operation of data. In order to mark out depend on data and need to cancel operation, through the analysis of components in log file, get corresponding rely on information, and use the restore components to produce compensation process, thereby eliminating PTP the effects of the operation.

3 A Integrity Recovery Algorithm for Malicious Partially Transaction

According to E7 rule, once PTP is malicious transaction, the transaction itself and data can be recovered by dependency relationship. This section focuses on the file system

operation, a recovery algorithm is given in terms of malicious partially-formed transaction after damage to the file. All the raw data and operations that depend on the malicious transaction can be found by the recovery algorithm. Because of the high cost of the former (redo) recovery method[12], we must make clear the execution semantics of each transaction, so we adopt the backward recovery (undo) method, and the file system will be restored to the previous consistent state.

3.1 Operation Event, Log and Dependency

Operation event is defined as an atomic operation of the conversion process. In the operating system, the conversion process can be viewed as a process, and the operation of creation, read, and write files can be viewed as an operational event of conversion process. Assuming log L records all operational events occur in the system E. An event in an audit log that represents a line or a record in an audit file, namely the log entries (entry). Such as a network data packet in the tcpdump log file. Each log entry is composed of one or more log elements, such as time, operation, and object, *etc*. In order to log analysis and recovery, the log entries include at least 2 basic elements: <time, subject, op, object, param>, that is the time, subject, operation, object and other related parameters.

Assuming that log record of events occurring sequence reflects the actual sequence of events. Chronological relationship using $\prec$ said.

Definition 1. Operation dependence. If the write operation w at time t_2 using data from the read operation r in t_1, $t_1 \prec t_2$, then write w relies on the read operation r, Denoted by $r \rightarrow_{op} w$.

Definition 2. Data dependence. If d_2 of write operation at the time t_2 depends on d_1 of the read operation at time t_1, $t_1 \prec t_2$, then data d_2 is dependent on data d_1, Denoted by $d_1 \leftarrow_d d_2$.

If PTP operations are malicious acts, by revoking its operation, to make the modified data back to a consistent state before the modification of purpose. The premise is that log records all events occurring in system, and stable storage must be saved enough to recover the data. In addition, since the PTP modification of the data is likely to be used by other PTP or TP, such as the existence of operational dependencies, we call this affected data for the "dirty" data. So, in addition to restore the operation of PTP but also to restore the "dirty" data of all operations.

Definition 3. If one of the following conditions is established, the "dirty" data will be generated:

1) the data of malicious transactions directly manipulate in a non-read mode.
2) the operation of malicious transactions data dependencies transactions in a non-read mode of operation.

In the operating system, there is a process to call the system call fork () to generate a new process. Malicious transactions can be directly derived from the new malicious transactions in this way. Similarly, according to dependencies of definition 1, read the malicious transaction to produce data of normal transaction, the transaction will also be affected.

Definition 4. If one of the following conditions is established, the transaction will be affected by the malicious transaction:

1) the transaction directly created by malicious transaction.
2) transactions that have a data dependency relationship to malicious transactions.

A dirty objects for transmission cycle is affected by the termination of its time. For the

purposes of the operating system in the process, namely dead or killed, in terms of the file to be deleted.

3.2 Recovery Policy and Algorithms

There are different recovery policies when taking the recovery action.One is a conservative policy that restores the correctness of the recovery and normal transactions, and can be used to postpone the execution of normal transactions by the recovery. The other is optimistic policy, the implementation of the recovery of the normal transactions, can only guarantee the correctness of the recovery, but because there is no complete recovery, and thus can not guarantee the correctness of normal transactions. There is also a radical policy, in the absence of a clear dependence on the situation of the implementation of a variety of transactions, may damage recovery and normal affairs, need to continue to re repair them. This policy will bring more cost of recovery, because it will have more work to do, more things to be destroyed. In this paper, we adopt the first two policies, and give the corresponding recovery algorithm.

Malicious transaction of object read operation does not affect the integrity of object, if the monitoring machine rejected the transaction on subject matter to write, create and delete operations, also will not affect the integrity of object. Therefore, in the design of algorithms and realize we will not restore these events.

Definition 4. Recovery goals.

1) the revocation of malicious transactions to the object of successful write, create and delete operations;
2) only revocation of malicious transactions for the operation of object;
3) remove all dirty data from the file system and recover it to the latest consistent state.

We first discuss the conservative recovery policy, when the system is restored for recovery operations in addition to other operations are stopped.

Algorithm is divided into two stages: Stage 1 is to define recovery point, that is, PTP is found to be first of the malicious transaction log records, and then from the point to start log file, to the end of file, to find data and transactions affected by the PTP malicious behavior, and to put the affected transactions in chronological order to resume log list *blacklist*; Stage 2, because there are some shared object file system, such as */dev/ttyv0*, its operation without recovery, it should first restore the list of shared objects operations and failed operations to remove, and then follow the first time after the order revoked *blacklist* operation malicious transactions.

Basic data structures and functions are defined as follows:

1) *tainted_ps*: represents malicious set of transactions; *tainted_objects*: represents an infected object set; *PS* represents set of transactions; *FS* indicates that the file object set.
2) list: the need to restore log list *blacklist*, which may be dependent on relationship of the list *tmplist*.
3) Log entry: e_i represents the i-th log entry.

Function definition: $op(e_i)$ represents operation in record (e_i); $sb(e_i)$ represents record (e_i) the subject or transaction;

4) represents record (e_i) the object.

Algorithm 1. Static recovery algorithm

Input: Recovery Point RP $\langle time_0, ptp_0, op_0, object_0, param_0 \rangle$

Output: It does not contain a consistent system state that is affected by PTP malicious behavior.

Step 1. Tracking process, to find all of the affected subjects and objects, and generate a

list of events needs to be restored.

1) Initialization

Blacklist first elements for the recovery point of the operation of the event〈 $time_0, ptp_0, op_0, object_0, param_0$ 〉,*tainted_ps* first elements is ptp_0 ; *tainted_objects* first element is $object_0$.

2) Starting from the RP, in the log files look forward log analysis.

```
repeat
if(tainted_ps.contains(sb(ei)) && blacklist.contains(ei)){
    switch(op(ei)){
        case w:
        case c:
            if(PS.contains(ob(ei)){
            /* Create a new transaction, and the new transaction is also a malicious
transaction */
                tainted_ps=tainted_ps.add(ob(ei));
            }else{
                /* The object is infected */
                tainted_objects=tainted_objects.add(ob(ei));
            }
            break;
    }
}else{
    switch(op(ei)){
        case r:
            if(tainted_objects.contains(ob(ei)){
                read(ei);
                tmplist=tmplist.remove(ei);
            }
            break;
        case w:
            if(tainted_ps.contains(sb(ei)) || tmplist.contains(ei)){
                write(ei);
                blacklist=blacklist.remove(ei);
                tainted_objects=tainted_objects.add(ob(ei));
            }else if(tainted_objects.contains(ob(ei))){
                /* Normal transactions on dirty data write operations */
                write(ei);
                blacklist=blacklist.remove(ei);
            }
            break;
        case c:
            if(tainted_ps.contains(ob(ei))){
                if(PS.contains(ob(ei))){
                /* Create new malicious transactions */
                tainted_ps=tainted_ps.add(ob(ei))}
            }

            if(FS.contains(ob(ei))){
                create(ei);
                blacklist=blacklist.remove(ei);
                tainted_objects=tainted_objects.add(ob(ei));
```



```
                }
                break;

            case d:
                if(tainted_ps.contains(sb(ei))){
                    if(FS.contains(ob(ei))){
                        blacklist=blacklist.remove(ei);
                    }
                }
                break;
        }
    }
    until{end of log}
```

Step 2. Reduction Process

1) removing the blacklist of the object to be read in the event;

2) removing the failed event blacklist in the shared object operate.

Step 3. recovery process

From the blacklist,in accordance with the order of time, revoked recovery.

Algorithm 2. Runtime recovery algorithm

Basic data structures and functions are defined as follows:

1) cleaned_objects means "being cleaned" a collection of objects and HSN;
2) the other is the same as algorithm 1.

Input: Recovery Point RP $\langle time_0, ptp_0, op_0, object_0, param_0 \rangle$

Output: It does not contain a consistent system state that is affected by PTP malicious behavior.

Step 1. Initialization and establish compensation matters.

1) Initialization

Blacklist first elements for the recovery point of the operation of the event<$time_0, ptp_0, op_0, object_0, param_0$>,*tainted_ps* first elements is ptp_0;

tainted_objects first element is $object_0$;*clean_objects* first element is empty.

2) Analysis of the affected data and operations, and the establishment of compensating transactions, submitted to the monitoring machine execution.

```
    repeat
    if(tainted_ps.contains(sb(ei))){
        switch(op(ei))}{
            case w:
                /* Establishment of compensating transactions CTP, and submitted to the monitoring machine */
                tainted_objects= tainted_objects.add(ob(ei));
                break;
            case c:
                /* Create new malicious transaction */
                if(PS.contains(ob(ei))){
                    tainted_ps=tainted_ps.add(ob(ei));
                }
                break;
        }
    }else{
        switch(op(ei)){
            case w:
```

```
                    if(!cleaned_objects.contains(ob(ei))){
                          /* tmp_objects illustrate the potentially infected object */
                          tmp_objects=tmp_objects.add(ob(ei));
                    }else if(w.HSN>ob(ei).HSV){
                          /* The object is removed before write operations, can cause the
object to become dirty */
                          tmp_objects=tmp_objects.add(ob(ei));
                          cleaned_objects=cleaned_objects.remove(ob(ei));
                    }

                    if(tmplist.contains(ob(ei))){
                          /* After reading an infected object write another object */
                          tainted_objects=tainted_objects.add(ob(ei));
                          /* The object moved from tmp_objects tainted_objects */
                          tmp_objects=tmp_objects.remove(ob(ei));
                    }
                    break;
              case r:
                    if(tainted_objects.contains(ob(ei))
                          || cleaned_objects.contains(ob(ei))
                          && r.HSN<=ob(ei).HSN){
                           /* After reading the operation, X is cleared */
                           read(ei);
                           tmplist=tmplist.remove(ei);
                    }
                    break;
              case d:
                    /* Can be deleted */
                    tmp_objects=tmp_objects.remove(ob(ei));
                    break;
        }
    }
    until{}
```

Step 2. Compensation transaction CTP after execution.

The *ob(e_i)* deleted from *tainted_objects*, the[*ob(e_i), ob(e_i).HSN*]was added to the *cleaned_objects* .

4. Restore Examples and System Design and Implementation

In the actual operation of the operating system files, the occurrence of the sequence of events as shown in Table 2, the dependence shown in Figure 2. Process A is a malicious process ,namely the existence of malicious acts PTP, needed to withdraw its action and the affected operating processes. Process A has writes to the file 1, so as to make the file 1 infected. Process B because of dependencies, also makes the file 2 infected. Process A through the system call fork() to create a new process C, this process also for malicious processes. Therefore, according to the algorithm 1, we will find the picture with black cord line produced illegal flow of information flow in the logo, need to cancel the sequence of events as [6,4,3,1].

Table 2. Occurrence Sequence of System Events

Time	Events
1	Process A writes File1
2	Process B reads File1
3	Process B write File2
4	Process A create(fork) Process C
5	Process C read File3
6	Process C write File4

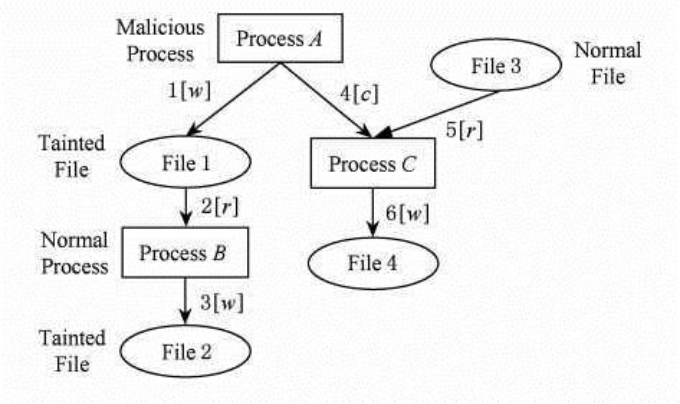

Figure. 2 Dependency Graph

Because of the file reads and failed operations without undo, so we put the contents of the audit log records into system calls, process and file three types of specific information as shown in Table 3. audit system can be used to achieve the specific BSM (basic security module). to determine the operations and processes and inter-process and inter-file dependencies, we analyze the system in FreeBSD 6.0 related system calls, based on the operation and their dependencies subdivided into table 4.

Table 3. Classifications and Contents of Log Records

Classification	Content
System call	ID,return value,return status
Process	ID,exit value,exit status
File	Access mode,path,file system,file name,argument

Table 4. Dependencies, Operations and System Calls

Dependence	Operation	System Call
Process→ Process	create	fork,vfork,clone,execve
	destory	kill,exit
	read	ptrace
	write	ptrace,kill
Process→ File	create	create,mkdir,link,mknod,pipe,sysmlink
	destory	unlink,rmdir,close
	read	read,readv,recv,access,stat,fstat,msgrcv
	write	write,writev,truncate,chdir,rmdir,chmod,chown, fchown,send,sendfile

5. Conclusion

Integrity of multi-policy is an important part of access control technology research in computer security. Trusted recovery is necessary for high-level security operating system. This paper presents a trustedrecovery monitoring model, which can solve some limits of strict security policy for access control. Firstly through the integration of IBAC(identify-based access control), TE(type enforcement) and RBAC(role-based access control), the use of compensatory well-formed transaction,we propose a recoverable integrity monitoring model, and the operation primitives for recovery are given. Secondly, combining the characteristics of a file system in operating system, this paper presents how to recover the file system to its last consistency secure state, in conservative and optimistic recovery policy respectively, by analyzing audit logs and undoing some malicious operations. This method can recover the system to a secure state in the face of failures and improves the availability of the system. This makes the monitor model recovery in the integrity of the system to ensure the proper implementation of the strategy at the same time, the compensating transaction for partial structure transaction recovery mechanism, enhance the ability of the system to tolerate malicious transactions, improve the usability of the system, which provides a reference for the trusted recovery mechanism of high level secure operating system.

Acknowledgments

This work is supported by 2013 National 863 Project (2013AA040302), National High Tech Research and development program (863 Program),and 2014 Shanghai economic and Information Commission project (ZB-ZBYZ-03-12-1067-1)

References

[1] D Bell, L LaPadula., "Secure computer systems: Mathematical foundations and model[R]".Technical Report M74- 244, The Mitre Corp, **(1973)**.
[2] J Biba Kenneth, "Integrity considerations for secure computer systems". Technical Report M74-244,The Mitre Corp, 1977
[3] D Clark,D Wilson. A comparison of commercial and military computer security policies [C] //Proc of the IEEE Symp on Security and Privacy. Los Alamitos:IEEE Computer Society,**(1987)**, pp. 55- 563
[4] L Badger,D F Sterne,D L Sherman,*et al*. Practical domain and type enforcement for UNIX [C] //Proc of IEEE Symp on Security and Privacy. Los Alamitos,CA:IEEE Computer Society,**(1995)** , pp.. 66-77.
[5] D F Ferraiolo,D R Kuhn. Role Based Access Control. 15th National Computer Security Conference. Oct 13-16, **(1992)**, pp. 554-563.
[6] P A Loscocco,S D Smalley.Integrating flexible support for security policies into the Linux operating syst em// Proceedings of the TREENIX Track :2001 USENIX Annual Technical Conference.Berkeley , CA , USA , **(2001),** pp. 29-42.
[7] J P Anderson. Computer security technology planning study,volume Ⅱ[R]. Bedford,MA: Electronic Systems Division,Air Force Systems Command,**(1972)**.
[8] D Povey. Optimistic security: A new access control paradigm [C] //Proc of the Workshop on New Security Paradigms. New York: ACM，**(1999)**, pp. 40 -45. [2012-12-11]. http:// portal, acm. org/citation. cfm?id= 335188.
[9] D Povey. Enforcing well-formed and partially formed transactions for UNIX [C] //Proc of the 8th USENIX Security Symp. Berkely: USENIX Association，**(1999)**, pp. 47-62.
[10] National Computer Security Center. A Guide to Understanding Trusted Recovery in Trusted Systems [R] . NCSC-TG-022 , Library No. 5236,061 , Version 1 , December **(1991)**.
[11] R S Sandhu. The typed access matrix model [C] //Proc of IEEE Symp on Security and Privacy. Los Alamitos,CA:IEEE Computer Society,**(1992)**, pp. 122-136.
[12] E N Elnozahy,L Alviri,Y M Wang,*et al*.A survey of rollback-recovery protocols in message passing systems [J]. ACM Computing Survey, vol. 34, no. 3, **(2002)**, pp. 375-408.
[13] D,Jiang, Z Xu, Z Chen, *et al*. Joint time-frequency sparse estimation of large-scale network traffic[J]. Computer Networks, 2011, 55(15): 3533-3547.Jinyu Hu, Zhiwei Gao and Weisen Pan. Multiangle Social Network Recommendation Algorithms and Similarity Network Evaluation[J]. Journal of Applied Mathematics, **(2013)**.

[14] J Hu and Z Gao. "Modules identification in gene positive networks of hepatocellular carcinoma using Pearson agglomerative method and Pearson cohesion coupling modularity [J]", Journal of Applied Mathematics, **(2012)**.
[15] Z Lv, A Tek, F Da Silva, *et al*. "Game on, science-how video game technology may help biologists tackle visualization challenges[J]", PloS one, **(2013),** vol. 8, no. 3, p. 57990.
[16] T Su, W Wang, Z Lv, *et al*. "Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve[J]", Computers & Graphics, **(2016)**, pp. 54: 65-74.
[17] S Zhou, L Mi, H Chen, Y Geng, "Building detection in Digital surface model", 2013 IEEE International Conference on Imaging Systems and Techniques (IST), Oct. **(2012)**.
[18] J He, Y Geng, K Pahlavan, "Toward Accurate Human Tracking: Modeling Time-of-Arrival for Wireless Wearable Sensors in Multipath Environment", IEEE Sensor Journal, vol. 14, no. 11, **(Nov)**, pp. 3996-4006.
[19] Z Lv, A Halawani, S Fen, *et al*. "Touch-less Interactive Augmented Reality Game on Vision Based Wearable Device [J]", Personal and Ubiquitous Computing, vol. 19, no. 3, **(2015)**, pp. 551-567.
[20] G Bao, L Mi, Y Geng, M Zhou, K Pahlavan, "A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract", 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. **(2014)**.
[21] D Zeng, Y Geng, "Content distribution mechanism in mobile P2P network", Journal of Networks, vol. 9, no. 5, Jan.**(2014)**, pp. 1229-1236.
[22] W Gu, Z Lv, M Hao. "Change detection method for remote sensing images based on an improved Markov random field [J]", Multimedia Tools and Applications, **(2015)**, pp. 1-16.
[23] Z Chen, W Huang, Z Lv. "Towards a face recognition method based on uncorrelated discriminant sparse preserving projection[J]", Multimedia Tools and Applications, **(2015)**, pp. 1-15.
[24] G Yan, Y Lv, Q Wang, Y Geng, "Routing algorithm based on delay rate in wireless cognitive radio network", Journal of Networks, vol. 9, no. 4, Jan. **(2014)**, pp. 948-955.
[25] Y Lin, J Yang, Z Lv, *et al*. "A Self-Assessment Stereo Capture Model Applicable to the Internet of Things [J]", Sensors, vol. 15, no. 8, **(2015)**, pp. 20925-20944.
[26] K Wang, X Zhou, T Li, *et al*. "Optimizing load balancing and data-locality with data-aware scheduling[C]" Big Data (Big Data), 2014 IEEE International Conference on. IEEE, **(2014)**, pp.119-128.
[27] L Zhang, B He, J Sun, *et al*. "Double Image Multi-Encryption Algorithm Based on Fractional Chaotic Time Series[J]". Journal of Computational and Theoretical Nanoscience, **(2015)**, pp. 12: 1-7.
[28] Su T, Lv Z, Gao S, *et al*. 3d seabed: 3d modeling and visualization platform for the seabed[C]. Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on. IEEE, 2014: 1-6.
[29] Y Geng, J Chen, R Fu, G Bao, K Pahlavan, "Enlighten wearable physiological monitoring systems: On-body rf characteristics based human motion classification using a support vector machine", IEEE transactions on mobile computing, vol. 1, no. 1, Apr. **(2015)**, pp.1-15.

