MA-IDS Architecture for Distributed Intrusion Detection using Mobile Agents

Chunsheng Li, Qingfeng Song, and Chengqi Zhang, *Senior Member, IEEE*

***Abstract*—Distributed intrusion detection systems (IDS) have many advantages such as scalability, subversion resistance, and graceful service degradation. However, there are some impediments when they are implemented. The mobile agent (MA) technology is of many features to suit the implementation of distributed IDS. In this paper, we propose a novel architecture --- MA-IDS with MA technology for distributed IDS. MA-IDS employs MA technology to coordinately process information from each monitored host, and then completes global information extraction of intruder actions. A prototype of mobile agent-based distributed intrusion detection system by following MA-IDS is developed. The system also introduces uncertain factor into intrusion decision, which accords with the objective reality that human behavior is changeful. We demonstrate the advantages and the potentials of MA-IDS by the result of evaluation.**

***Index Terms*—Distributed Intrusion Detection, Distributed System, Mobile Agent, Network Security**

I. INTRODUCTION

With the rapid growth of Internet, the security-relevant incidents have being increased. In addition, cracking technology has evolved into complex approach such as coordinated attack and cooperative attack. Under these circumstances, there is a great need for software tools that can automatically detect a variety of intrusions. As an important gatekeeper of network, the intrusion detection systems (IDS) must have the ability to detect and defend intrusions more proactively in shorter period. However, present-day IDS have some shortcomings as follows [2,7]:

- Most IDS detect attacks throughout an enterprise by analyzing information from a single host, or a single network interface, at many locations throughout the network. IDS components are lack of communication and cooperation each other. This limits the capability to detect large-scale distributed attacks.
- Most commercial IDS are built in hierarchical architecture, which is a tree structure with a control system at top, information aggregation units at the internal nodes, and sensor units at leaf nodes. In this kind of system, large amount of data transferred across the network may result in network congestion.
- Because of the reliance on hierarchical structures, many IDS are susceptible to be attacked. An attacker can cut off a control branch of the IDS by attacking an internal node or even decapitate the entire IDS. Typically, such critical components have been hardened to resist direct attack. Nevertheless, other survivability techniques such as redundancy, mobility, dynamic recovery etc. are lacking in current implementations.
- Many IDS cannot adequately combine history intrusive alarms to analyze future intrusive behaviors. "Knocking attack" is a good example. If IDS look upon multiple times of intrusions to different hosts from the same source at relative long interval as isolated events, it cannot find this kind of intrusion. It means that many IDS have no ability to dynamically adjust detective policy by the former intrusive results.

In order to solve aforementioned shortcomings of current IDS, we propose a novel architecture with mobile agent (MA) technology for distributed IDS. We call the architecture as MA-IDS for short. Software agent is a program that can exercise an individual or organization's authority, work autonomously toward a goal, and meet and interact with other agents [2]. Mobile agent is a particular type of software agents, which has the capability of moving from one host to another. Mobile agent is of the features of reducing network overload, overcoming network latency, synchronous and autonomous execution, robustness and fault-tolerance, system scalability, and operating in heterogeneous environments. To this end, MA technology is very suitable to solve intrusion detection in a distributed environment [4].

MA-IDS employs MA technology to coordinately process information from each monitored host, and completes global information extraction of intruder actions. We implement a prototype of mobile agent-based distributed intrusion detection system by following MA-IDS. The system also introduces uncertain factor into intrusion decision, which accords with the objective reality that human behavior is changeful.

The remaining sections of this paper are organized as follows. Section 2 discusses the state of the art about IDS and IDS with MA technology. Section 3 describes our approach and reports in details how the components work in the MA-IDS. Section 4 outlines the implementation of MA platform in our

C. Li is now with the Faculty of Information Technology, University of technology, Sydney, BROADWAY, NSW 2007 Australia (telephone: 61-2-9514 4534, e-mail: csli@it.uts.edu.au). He is also with the School of Computer Science and Engineering, Daqing Petroleum Institute, Daqing, Heilongjiang 163318 China.

Q. Song is with Beijing Venus Info Tech Inc. 188#, No.12 ZhongGuanCun Southern Street, HaiDian District, Beijing 100081 China (e-mail: song_wanru@sina.com.cn).

C. Zhang is with the Faculty of Information Technology, University of technology, Sydney, BROADWAY, NSW 2007 Australia (telephone: 61-2-9514 7941, e-mail: chengqi@it.uts.edu.au).

ICITA2004 ISBN 0-646-42313-4

system based on an existing MA platform Gypsy. Section 5 shows a quantitative evaluation of our architecture according to the prototype. Section 6 discusses advantages and drawbacks of our architecture and presents conclusions.

II. RELATED WORK

The research of IDS is classified into three levels: approach level, implementation level, and evaluation level. In approach level, intrusion detection systems are typically categorized between the misuse detection approach and the anomaly detection approach, as well as between the centralized systems and the distributed systems [4]. Balasubramaniyan et al. proposed an agent-based architecture for a distributed intrusion detection system based on multiple independent entities working collectively [2], but the low-level implementation and communication mechanisms are still open. Kachirski and Guha researched intrusion detection using mobile agents in wireless ad hoc networks [6]. In implementation level, Hoagland and Staniford desined the user interface to an intrusion detection system console [5]. They implemented web-based console for examining large files of alerts from the open source IDS Snort. But they didn't fully finish the implementation. In evaluation level, Amoroso et al. proposed a set of criteria for comparing and assessing intrusion detection systems [1]. The theory, requirements classes, metrics, and practical application of the criteria were discussed. Birch did the technical evaluation of rapid development and re-deployable intrusion detection systems [3]. All above work promoted the research on IDS.

Implementation of intrusion detection systems with mobile agent technology is one of the new paradigms for intrusion detection. The research is being conducted at a number of research labs [2,4,6]. However, just a few of results have been published. At present, the research on distributed IDS with MA technology is rapidly growing. The Intrusion Detection Agent (IDA) system developed by The Information-technology Promotion Agency in Japan is a hierarchical intrusion detection system with intelligence gathering sensors at the leaves and a central manager at the root. The Java Agents for Meta-Learning (JAM) project at Columbia University applied meta-learning to distributed data mining using intelligent agents. It used intelligent, distributed Java agents to learn models of fraud and intrusive behavior. MAIDS (Mobile Agent Intrusion Detection System) was developed by Iowa State University is a distributed IDS based on MA technology. It build a model for an intrusion activity with Software Fault Tree Analysis (SFTA), and transform the SFT model into Intrusion Detection model by the use of Colored Petri Net(CPN).

MA may enhance the performance of IDS and even offer IDS some new capabilities, however, these benefits is not easy obtained. We could learn from these existing systems that there are three main research areas in IDS with MA technology:

- Performance enhancements: take advantage of mobility and autonomy of MA to obtain better performance.
- Design improvements: use MA technology to enable novel paradigms for detecting attacks.
- Response improvements: use MA technology to enable novel and efficient automated responses to attacks.

III. MA-IDS ARCHITECTURE

The MA-IDS architecture chiefly includes four components: Manager, Assistant MA, Response MA, and Host Monitor Agent as shown in Fig. 1.

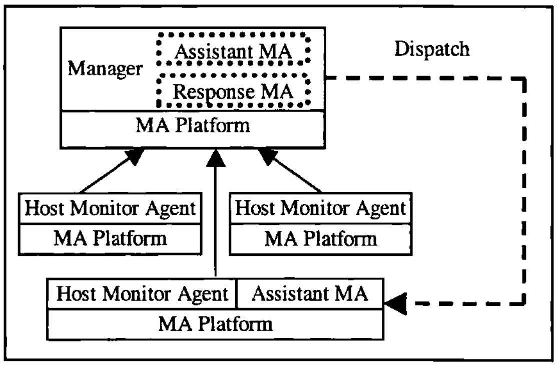


Fig. 1. System Architecture — MA-IDS

Each monitored host in the network is installed with a Host Monitor Agent. The Host Monitor Agent can cooperate several detection subagents to complete local intrusion detection function. If the intrusion can be determined at certain monitored host, the Host Monitor Agent reports the intrusion directly to Manager. Otherwise the Host Monitor Agent asks Manager for aid and it only records the suspicious activity. When Manager receives the assistant request, it will dispatch an Assistant MA to patrol in the network to other monitored hosts to gather information, for determining whether some suspicious activities in different hosts can be combined to be a distributed intrusion. Upon the Assistant MA's return, Manager analyzes the gathered information then makes the distributed intrusion identification. If a distributed intrusion is found, Manager will possibly dispatch a Response MA to do intelligent response to each monitored host.

For the purpose of the flexibility and security of the system, we wrap most components into mobile agents. Consequently, each host must be installed a MA Platform that can provide runtime environment for migration and execution of MA.

A. *Host Monitor Agent*

In Host Monitor Agent, there are three subagents responsible for monitoring network connection, file operation and privilege operation, respectively, as shown in Fig. 2. They are called network detection subagent, file detection subagent and user detection subagent. Intrusion Analyzer computes the suspicion level of certain access according to suspicion levels of its activities monitored by those three subagents and the suspicion levels of source host and user. If the suspicion level of certain access rises high enough (the threshold is 0.8 by case study in this system), the access will be thought as an intrusion and directly reported to Manager. If the access cannot be determined to be an intrusion, Host Monitor Agent will submit an aid request to Manager.

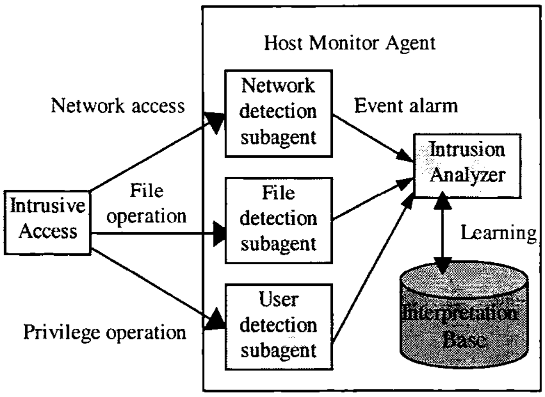


Fig. 2. Host Monitor Agent structure

There are many technologies that can detect local intrusion, such as expert system, artificial intelligence, pattern matching, however these technologies have some known drawbacks. For instance, some technologies (expert system) need numerous initial training and maintaining, which cannot be updated easily after the change of user security policy. Other technologies (pattern matching) are dependent of full-scale patterns, which cannot identify intrusions if patterns are not enough. In view of the characteristics of local intrusion detection technologies above, we propose a synthetic fuzzy analyzing method.

Intrusion detection systems are the tools that seek to detect attacks against computer systems by monitoring the behavior of users, networks, or computer systems. However, it is a tough task to distinguish intrusion activities and normal activities, and future attackers may use completely unknown patterns that are unexpected and difficult to detect. We add the concept of "attempt" to the intrusion definition. An intrusion is defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource".

How do we identify the intrusive attempt? Our approach is to endow each monitored entity or operation a "suspicion level", so that the intrusive attempt can be determined according to the monitored entity itself and its activities. In the MD-IDS architecture, each activity is specified with a suspicion level and the suspicion level increases with the occurring times of the activity. For instance, the suspicion level set for "port scanning" is {0/0, 0.4/1, 0.6/2, 0.9/3, 1/4, 1/5...}, and the suspicion level set for "login failing" is {0/1, 0.25/2, 0.5/3, 0.75/4, 1/5, 1/6...}. Each host that is ever connected to the network is assigned a suspicion level. These suspicion levels are stored into the Interpretation Base, which can be changed with the analyzing results of the Intrusion Analyzer about intrusion increases (This process is called "Learning"). When an intrusive access is determined, the suspicion levels of its source host and its user will be increased.

B. Manager

In the MA-IDS architecture, Manager is the center for controlling and coordinating all other components. Manager maintains configuration information about all components including Host Monitor Agents, MA Platforms, Assistant MA and Response MA. When one of the components mentioned above is added to the system, it must register in the component list of Manager. Manager is also responsible for creating, dispatching, accepting and removing mobile agents according to the host requests and environment. In addition, Manager is responsible for detecting the security of any mobile agent in the system. Finally, Manager receives intrusion alarms from Host Monitor Agents and the information gathered by Assistant MAs, and performs global intrusion response by Response MAs.

Manager mainly consists of five modules: global agent management module, global agent security module, communication module, distributed intrusion analysis module, and intrusion response module. The global agent management module is responsible for system-wide management functions for all agents and MA Platforms in the system. This module creates agents, assigns each agent a unique identifier (AID), and determines the itinerary of mobile agents. The module maintains a MA directory, where locations and AIDs of mobile agents are kept. The track of any mobile agent will be recorded in the MA directory with the mobile agent migration between the hosts. This module also maintains a timer for each dispatched mobile agent for determining whether the mobile agent has been lost or destroyed.

The global agent security module is responsible for detecting violations of agent integrity and ensuring the availability of all dispatched agents. After a mobile agent returns to Manager, the module will analyze the mobile agent to determine whether some security violations have occurred. The security module queries the status of each MA platform. If any MA Platform on the itinerary is unavailable, the module may perform further analysis to determine the causes. The key management of all public keys in the trust architecture of the system is completed in Manager. Manager serves as the certification authority for the whole system and determines when new keys should be generated and old keys revoked.

The communication module is responsible for all remote interactions between Manager and all other system components, including communications with MA Platforms and Host Monitor Agents.

The distributed intrusion analysis module uses the similar method to the Host Monitor Agent for identifying distributed intrusions. It maintains an access list for each suspicious access to the network but certain host, which includes source host IP and its suspicion level, login user and its suspicion level, destination host IP and current access suspicion level. When the Assistant MA surveying suspicious network access returns, the gathered information will be added into this access list. The item in access list is not removed unless the non-access interval of this item is long enough.

The intrusion response module is responsible for making response decision and dispatching Response MAs with the response to relevant monitored hosts. For example, if Manager received a local attack reported by a Host Monitor Agent or determined a distributed attack, it would dispatch a Response MA to the host to enhance the suspicion level of this host. If the host did not access the network for a long time, the suspicion level of this host might be reduced upon receiving the notification of a Response MA.

C. *Mobile agents*

There are two kinds of agents in MA-IDS architecture: fixed agents and mobile agents. Fixed agents include Manager and Host Monitor Agent, each of which completes its tasks in a specific host. Mobile agents are the critical components in the MA-IDS architecture and include Assistant MA and Response MA. They are dispatched by Manager and patrol along assigned monitored hosts to carry out specific tasks. These two types of agents have the similar structure, however, there are some differences between them.

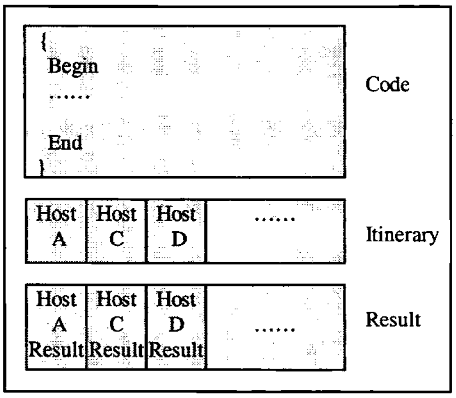


Fig. 3. Mobile Agent Structure

The mobile agent is composed of three parts: code, itinerary, and result as shown in Fig. 3. Those Agents follow a predetermined itinerary established by Manager. Upon arrival at a host, the MA Platform starts to execute the code. The data resulted from the agent execution on each host is stored in the result area, which is divided into multiple blocks for each host in the itinerary. The patrolling MA and Manager maintain the same timer together. If its timer decreases to zero, the patrolling MA must return to Manager whether it has completed its task or not. If Manager finds the patrolling MA cannot come back after its timer becomes to zero, it will analyze the cause and dispatch another mobile agent again. By setting timer for patrolling MA, a missed agent can be detected and regenerated with minimal data loss.

Although Manager and Host Monitor Agents are fixed in the specific hosts, they also can move in the network according to the environment. Because Manager and Host Monitor Agents both have large sizes, it is difficult to move them frequently in the network. However, the mobility still can bring two advantages: flexibility and security. Firstly, Manager can be installed in any operating system as long as it supports JVM (Java Virtual Machine), and Host Monitor Agents can be casually installed and updated. Secondly, MA platform can provide accident restoration function. If there is any intrusion interfering the execution of a mobile agent, this agent can be restored by stored data and status information, or even move to another MA platform. This type of agent only includes two parts: code and result, but without itinerary part.

IV. Implementation of MA Platform

We employ the mobile agent platform Gypsy developed at the Technical University Vienna because it satisfies the development needs of an intrusion detection system. Gypsy is a flexible and dynamically extensible platform for experimenting with mobile agent programming. Gypsy adopted Java security mechanisms for secure class loading. Furthermore, it employed Java Sandbox security model including code signing and class loading. The existing Gypsy platform have not sufficient features for our architecture, so we must enrich the platform by adding some new parts and making some changes about Gypsy platform.

Because mobile agents migrate in a potentially threatening environment, they must be protected from major threats. For agent migrating security, the confidentiality, authentication and integrity of agents during network transmission must be ensured. Communication content between MA platforms cannot be eavesdropped to reveal. Sender and receiver can identify whether the other party is legal. Any tampering of agents in the process of network transmission must be detected as quickly as possible. For agent execution security, because agents must execute in plaintext form, they may be vulnerable to eavesdroppers on the host while executing. Unfortunately, no effective solution has been found so far to protect executing agent code.

Gypsy platform has not security function. In order to achieve the three goals of aforementioned agent transmission security, we develop an agent transmission security module. The module is designed to build on the benefits of public-key cryptography, symmetric-key cryptography, and message authentication codes (MAC). Firstly, each agent is encrypted before it is transported between hosts. Agents are encrypted using a symmetric key algorithm with a one-time session key. Then, this session key is encrypted using a public key algorithm, implementing the authentication of the other side to the originator. Finally, the MAC about important contents of the agent is computed for receiver to verify the integrity of the agent upon the arrival.

Another new module we have developed is the agent resuscitation module, which is responsible for aiding the regeneration and recovery of mobile agents. There are three cases in which agents need to be resuscitated. The first one is that agent may lose during migration. The second one is that agent may suffer malicious attack or deletion. The last one is that the MA Platform may not work in order. In this module, mobile agents are stored with their data and status at regular intervals. If an accident occurs, the agent resuscitation module will try to regenerate an agent with the latest stored data and status information, then activate it.

V. Evaluation

We develop a prototype by using the MA-IDS architecture for evaluating the performance of the MA-IDS-based system. The experimental environment consists of four Sun workstations (Ultra5 and Ultra10).

A. *Intrusion detection ability evaluation*

Intrusion detection in this system includes two types: local intrusion detection and distributed intrusion detection. Their detective abilities indicate the core function of the whole system. In order to test the validity of local intrusion detection, we select some cracking tools aimed at local attacks and distributed attacks, and simulate some attacks by using those tools. We divide local attacks into several classes, which could basically involve in popular attack approaches. Those attack types are unauthorized access, buffer overflow, password guessing, rootshell execute, file operation, port scan, privilege violation, etc. The result is that 94.1% attacks can be detected.

B. *System performance evaluation*

Local intrusion detection may usually consume some system resource, so the system resource use is an important metric for evaluating a local intrusion detection tool. Many experiments show that the increase of resource (CPU, memory etc.) use of the monitored host is very light after running the Host Monitor Agent so that the user cannot perceive easily. Generally, the use time of CPU is less than 1%, and approximate 5% memory is exhausted. The detail data is related to the CPU performance, memory capacity, operating system, current process number, etc.

As such, the additive network traffic is one of main factors for evaluating a distributed intrusion detection system. In this system, Host Monitor Agents and Mobile agents all take the merit of lessening network traffic. The network traffic generated by this system is less than 100k bytes per second.

C. *Mobile agent performance evaluation*

As we know, the size of mobile agent will affect the speed of its migration. Big mobile agents may cause the system performance degradation. In our system, both Assistant MA and Response MA are less than 2k bytes. We measured the interval from the departure of Host Monitor Agent or Manager to their return. It took a long time that agents migrated with authenticating and encrypting, though the transportation of those agents was very fast. The results are shown in Table 1.

Table 1. Interval of Agent Round Trips

Number of Mach Round Trip Time (sec)	1	2	3	4
Assistant MA	6.342	10.683	15.215	23.427
Response MA	5.654	9.987	13.860	21.754

VI. Conclusion

MA technology is efficient for enhancing security, flexibility and cooperative detective ability of distributed IDS. Our prototype system shows the following advantages of the MA-IDS architecture:

- The IDS can keep running even if the failure of some agents. Moreover, Mobile agents in this system are capable of evading attackers and resurrecting themselves if they are attacked.
- The mobility of MA makes it possible that distributed intrusion can be detected by means of data correlation and cooperative detection. Assistant MA can correlate all suspicious events occurred in different monitored hosts. They can provide more accurate alarms. Response MA can dynamically increase/reduce the suspicion level of certain host or login user. They can enhance the reactive capability of intrusion detection result.

However, MA-IDS architecture still has some problems. If the location of Manager were found by attackers, the IDS would be in a dangerous situation. Another problem is that mobile agents may spend more time. Although the implemented prototype proves that the architecture is available, some implementation must be improved.

References

[1] E. Amoroso, and R. Kwapniewski, "A selection criteria for intrusion detection systems," in Proceedings of 14th Annual Computer Security Applications Conference, Phoenix, USA, Dec. 1998, pp.280-288.

[2] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, "An architecture for intrusion detection using autonomous agents," in Proceedings of 14th Annual Magnetism Computer Security Applications Conference, Phoenix, USA, Dec. 1998, pp. 13-24.

[3] A. Birch, "Technical evaluation of rapid deployment and re-deployable intrusion detection systems (RDIDS/RIDS)," in Proceedings of Institute of Electrical and Electronics Engineers 1992 International Carnahan Conference on Security Technology, Atlanta, USA, Oct. 1992, pp. 34-40.

[4] P.C. Chan and V.K. Wei, "Preemptive distributed intrusion detection using mobile agents," in Proceedings of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Jun. 2002, pp. 103 -108.

[5] J.A. Hoagland and S. Staniford, "Viewing IDS alerts: lessons from SnortSnarf," in Proceedings of DARPA Information Survivability Conference & Exposition II, Vol 1, Anaheim, USA, Jun. 2001, pp.374-386.

[6] O. Kachirski and R.Guha, "Intrusion detection using mobile agents in wireless ad hoc networks," in Proceedings of IEEE Workshop on Knowledge Media Networking, Jul. 2002, pp.153-158.

[7] N. McAuliffe, D. Wolcott, L. Schaefer, N. Kelem, B. Hubbard, and T. Haley, " Is your computer being misused? A survey of current intrusion detection system technology," in Proceedings of the Sixth Annual Computer Security Applications Conference, Tucson, USA, Dec. 1990, pp.260-272.

Second International Conference on Information Technology & Applications

Conference Organising Committee

Conference Chair Prof Shi Guangfan Heilongjiang University, China

Conference Vice-Chair Dr. Fu Yuzuo Shanghai Jiao-tong University, China

Techical Chair Dr. Sean He University of Tech Sydney, Australia

International Advisor Dr Dapeng Tien Charles Sturt University, Australia

ICITA 2004 is organised by:

Heilongjiang University, Harbin, China

http://www.hlju.edu.cn

And supported by:

Shanghai Jiao Tong University, Shanghai, China

http://www.sjtu.edu.cn

IEEE, NSW Section, Australia

http://ewh.ieee.org/r10/nsw/

IEEE, CS Chapter, Beijing, China

http://www.cie-china.org/ieee-beijing/

Saora Inc., Japan

http://www.saora.com/

ICITA 2004 Harbin, China http://www.icita.org

Second International Conference on Information Technology & Applications

ICITA 2004

Harbin, China

8-11 January 2004

http://www.icita.org

[illegible], China

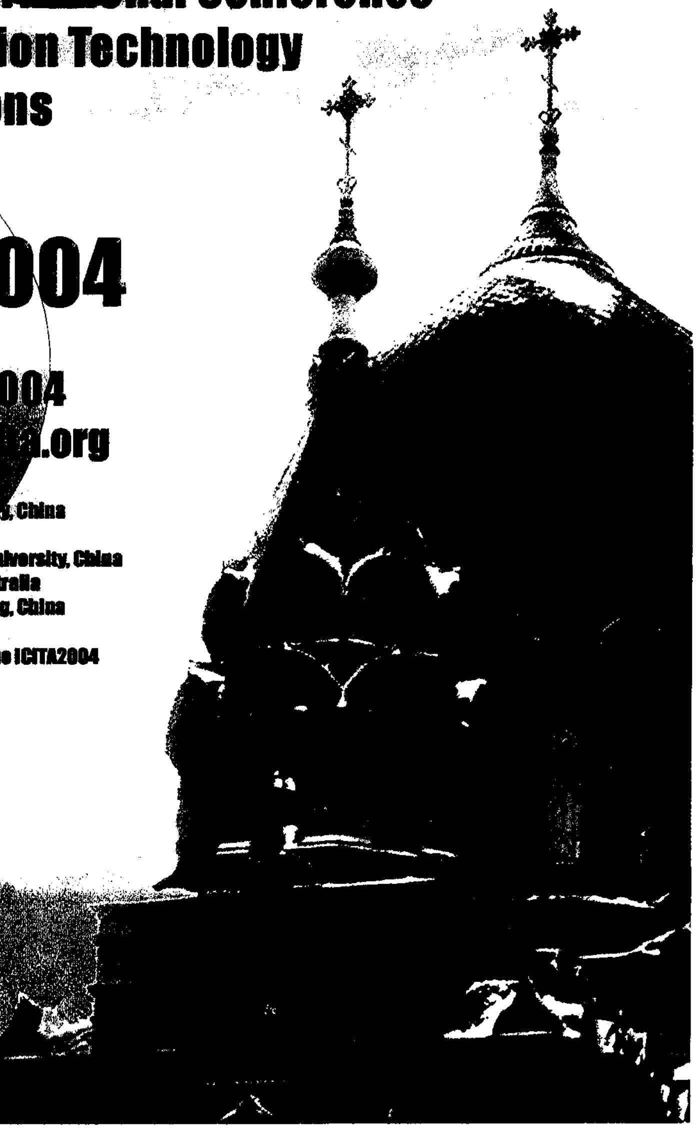
[illegible] University, China

[illegible] Section, Australia

IEEE, CS Chapter, Beijing, China

Saora Inc. , Japan

papers published at the ICITA2004 will be EI indexed.



Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)

Contents

Theme 1: IT in Telecommunication and Mobile Communications

Theme 2: IT in Multimedia; Computer Networking; and Database Interface

Theme 3: IT in Image Compression.

Theme 4: Web Content Generation, Usage and Management.

Theme 5: IT in Engineering: AI, Signal/Image Processing; Power & Power Electronics; Sensors.

Theme 6: IT in Education and Open Learning.

Theme 7: IT in Health, Medical Care and Biomedical Engineering.

Theme 8: Security and Hacking.

Theme 9: E-commerce.

Theme 10: IT in Animation and Graphic Design.

Theme 11: IT in the Humanities and Social Science.