논문 2011-48CI-4-7

클라우드 컴퓨팅 환경을 위한 퍼지 논리 기반 상황인식 접근 제어 모델

(Fuzzy Logic-based Context-Aware Access Control Model for the Cloud Computing Environment)

징스다[*], 정 목 동[**]

(Jing Si Da and Mokdong Chung)

요 약

무선 환경에서의 인증 모델은 많은 보안 위험을 내포하고 있다. 그러나 이 분야에 일반적으로 사용되는 표준 기술이 존재하는 것이 아니어서 본 논문에서는 무선 인증 환경에서 보다 강화된 보안을 제시할 수 있는 퍼지 논리 기반의 인증 모델을 제시한다. 제안하는 모델을 위해서 퍼지 논리 기반의 분류 방법을 사용하고 적절한 판단을 위해서 개선된AHP 알고리즘과 사례 기반의 추론 방법을 추가적으로 사용한다. 개선된AHP 알고리즘을 사용해서 다양한 상황정보를 수치화하고, 제안된 인증 모델을 사용해서 입력 데이터에 대한 보안 등급을 계산한 다음, 다양한 환경의 상황정보를 가진 무선환경에 제안된 인증 모델을 안정적으로 사용할 수 있을 것이다. 향후에는 상황인식 연구를 클라우드 컴퓨팅 분야를 포함한 다양한 분야로 제안 모델을 확대하면 더욱 안전한 보안 환경을 기대할 수 있을 것이다.

Abstract

Authentication model in the wireless environment has many security vulnerabilities. However, there is no adapting standard method in this field. Therefore, we propose a fuzzy logic based authentication model to enhance the security level in the authentication environment. We use fuzzy logic based classification to construct our model, and also additionally utilize improved AHP and case-based reasoning for an appropriate decision making. We compute the context information by using the improved AHP method, use the proposed model to compute the security level for the input data, and securely apply the proposed model to the wireless environment which has diverse context information. We look forward to better security model including cloud computing by extending the proposed method in the future.

Keywords : access control, wireless authentication, improved AHP method, cloud computing

Ⅰ. Introduction

Context management is a dynamic computer process that uses 'subjects' of data in an application. Authentication model in the wireless environment has many security vulnerabilities. However, there is no adapting standard method in this field. We use fuzzy logic based classification to construct our access control model, also we additionally utilize improved

[*] 학생회원, [**] 정회원-교신저자, 부경대학교 컴퓨터공학과
(Dept of Computer Eng., Pukyong National University)

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0024053).

접수일자: 2011년1월17일, 수정완료일: 2011년7월5일

AHP(Analytic Hierarchy Process) and case-based reasoning[1] for an appropriate decision making. The characteristics of the proposed model in this paper are as follows: these methods are useful in context classification, the advantage of fuzzy logic is that it can classify the context information properly, and the advantage of improved AHP is that it can make decisions by some figures and settings. We try to apply the proposed model in the diverse domains, such as SaaS[2], PaaS[3], and other Cloud Computing[4] model, wireless threatening area[5], and fuzzy algorithm related arena[6]. Fuzzy logic based context analysis module provides functionality to the whole system so that it can be integrated in authentication environment. The security of the whole system will be improved by using the proposed model. This paper is constructed as follows. In section Ⅱ, we summarize the related work. In section Ⅲ, we design and implement fuzzy logic based context analysis module by using C++ and improved AHP method. Section Ⅳ illustrates the result of the proposed model, CAAC (Context-Aware Access Control). Section Ⅴ concludes this paper with the further study.

Ⅱ. Related Work

1. Access Control Models and Techniques

Access control models are suggested to enforce the rules and objectives of an established security policy and to dictate how subjects can access objects. There are three models that will be covered in this section: discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC).

(1) DAC

A discretionary access control (DAC) model allows the owners of objects (resources) to control who accesses them and what operations can be performed on the objects. This is typically done through access control lists (ACLs), where permission is granted on a need-to-know basis.

(2) MAC

Mandatory access control (MAC) models do not leave access decisions up to the data owner, instead systems compare the subjects' clearances and need-to-know to the objects' classification to either grant or disallow access. Every object has a security level assigned to it, which includes classification information (top secret, secret, etc.).

(3) RBAC

Role-based access control (RBAC) models, also called nondiscretionary models, make access decisions based on the rights and permissions assigned to a role or group, not an individual user. Administrators create roles, or groups, which act as containers for users[7].

2. Typical Usage of Cloud Computing

This use case involves multiple clouds working together, including both public and private clouds. A hybrid cloud[8] can be delivered by a federated cloud provider that combines its own resources with those of other providers. A broker can also deliver a hybrid cloud; the difference is that a broker does not have any cloud resources of its own. The provider of the hybrid cloud as shown in Fig. 1, must manage cloud resources based on the consumer's terms[9].

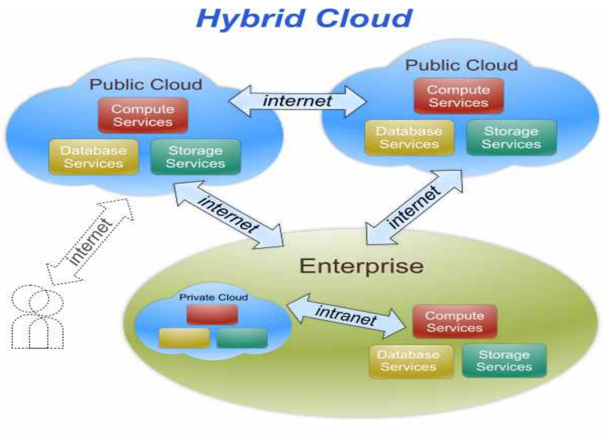


그림 1. 클라우드 컴퓨팅 응용 사례
Fig. 1. Cloud Computing Use Case[9].

3. Analytic Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP) is a structured technique for dealing with complex decisions. Rather than prescribing a “correct” decision, the AHP helps the decision makers find the one that best suits their needs and their understanding of the problem. The AHP steps are described as following :

Step 1: Build “hierarchy”

a) Identify the choices you're considering.

b) Outline the major factors you'll use to evaluate each option.

c) Identify criteria (and any sub-criteria of these) that you need to consider for each of these major factors. Link these to the major factors.

d) Continue to build a hierarchy of decision criteria until all factors are identified and linked.

Step 2: Establish priorities

e) Using paired comparison, determine your criteria preferences (perhaps A is a little more preferred to B, B is much more preferred to C, and so on).

f) Rate these preferences from 1-9.

g) Repeat this for each level in your hierarchy.

Step 3: Synthesize, or combine, the ratings

Calculate weighted criteria scores that combine all of the ranking data.

Step 4: Compare the alternatives

Using those combined scores, calculate a final score for each alternative[1].

4. Improved AHP Description

The effect of Behavioral context is more than General context, that's min {History, Status} ≥ max {Location, Login time}, therefore we can compute this function with one parameter. General context includes location and login time. Behavioral context includes history and status. We can compute this function with one parameter. Fig. 2 shows that we can compute the value of similarity and changing frequency, and we compare with them, then we get the weighting factor such as similarity is 0.6 in location and login time, and it's 0.7 in status and history. We set ‘location and login time’ as A, and ‘status and history’ is B. min{History, Status} ≥ max{Location, Logintime}, A*0.4 ≥ 0.7(1-X), X ≥ 7/11. We choose X=7/11. The weight value of ‘Status and History’ is 0.63, and the value of ‘Location and Logintime’ is 0.37, we choose the Average/Priority Vector, thus the weight value of ‘Status and History’ is 0.7, the value of ‘Location and Login time’ is 0.3.

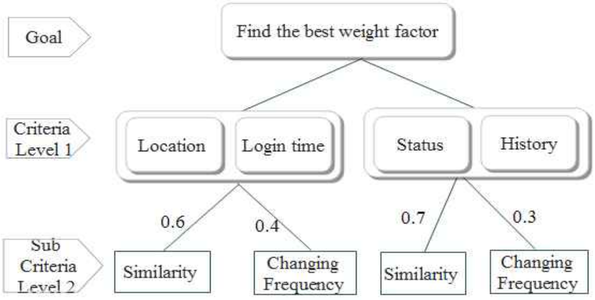


그림 2. 개선된 AHP 분석 방법
Fig. 2. Improved AHP Analysis Method.

5. Fuzzy Logic based Classification

Fuzzy algorithm is a method of clustering which allows one piece of data to belong to two or more clusters[10~11]. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. Therefore, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the

표 1. 퍼지K-평균수 분류알고리즘
Table 1. Fuzzy K-means Classification[12].

Algorithm: Fuzzy K-means Classification
1. ***Place*** *K points* ***into*** *the space represented by the objects that are being clustered. These points represent initial group centroids.*
2. ***Assign*** *each object* ***to*** *the group that has the closest centroid.*
3. *When all objects have been assigned,* ***re-calculate*** *the positions of the K centroids.*
4. ***Repeat*** *Steps 2 and 3* ***until*** *the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, we may notice that the k centroids change their locations step by step until no more changes are done. In other words centroids do not move any more[12]. Table 1 shows fuzzy K-means classification.

Ⅲ. Design and Implementation of CAAC

1. Overview of CAAC

In this paper, we presented the Context-Aware Access Control (CAAC) model that provides context aware access control for pervasive applications. The operation of the model is illustrated using a sample application scenario. Compared to the traditional access control mechanisms, the CAAC model can provide improved security for pervasive applications. However, access control alone is not sufficient and CAAC must be combined with feasible authentication mechanisms to secure pervasive applications in the real world. Fig. 3 shows the first level of CAAC model, which includes three sub modules:

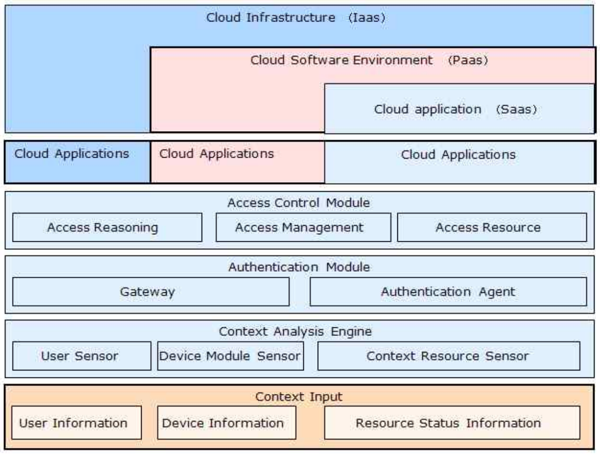


그림 3. 제안된 보안 모델
Fig. 3. Proposed Security Model.

(1) Sub Modules of CAAC

Context Engine :

The function of Context Engine is providing learners with knowledge skills to learn context environment

Authentication module :

Authentication module is used to check the input values and grant principles to it.

Access control module :

Access control module is used to control the inner service. There are some major components, such as context resource and access management.

2. Algorithm of CAAC

(1) Definition of Context

We can divide context information into diverse categories. Context information description has four types as follows:

Definition 1. Resource Status Information

Case 1: Location

When the user logins, for instance from Busan, and changes his location frequently, the score is low.

Case 2. History

History is user's login memory. An access control list table is constructed when a user logins. If the login information is much different from the last, the score is low.

Case 3. Status

Status refers to the phase stability as the user logged in. The higher frequency he changes his login status, the less score he will gain.

Case 4. Login time

Login time means the period that the user is on line. The longer he is on line, the more scores he gets.

Case 5. Changing frequency

Changing frequency refers to how often context information changes, measured by the ratio between absolute value of cases and time. In location case, the absolute value means times.

Definition 2. Device Module Information

The WAP browser requests an XHTML MP page from the server. The server receives the request and delivers the XHTML MP document to the WAP browser.

Case 1: Wireless Module

Wireless module permits services, such as long range communications, that are impossible or impractical to have wired implementation. The term is commonly used in the telecommunications industry to refer to telecommunications systems (e.g. radio transmitters and receivers, remote controls, computer networks, network terminals, etc.) Table 2 shows Device Module Algorithm.

Case 2: PC Module

The profile header holds the URL to the UAProf (User Agent Profile) document of the wireless device. The UAProf document contains information about the wireless device's characteristics and capabilities such as screen size, supported character sets, number of softkeys, etc. The details of the profile header and the UAProf document are out of the scope of this section and it will be covered in the later paper.

표 2. 단말 모듈 알고리즘
Table 2. Device Module Algorithm.

```
   Algorithm : Device Module
 static booleanchoose(StringuserAgent){
   if (userAgent.indexOf(“Noki”) > -1 ||
   userAgent.indexOf(“Eric”)   > -1 ||
   …
   userAgent.indexOf(“Wapa”)  > -1 ||
     userAgent.indexOf(“Oper”)  > -1)  {
     return true;
     } else {
       return false;
     }
```

Definition 3. User Information

Case 1: Same User (Personal Account)

Face recognition is the process of automatically determining whether two faces are the same person. A number of factors make this a challenging problem for computers. Faces in images and video can be captured at various resolutions, quality, and lighting conditions. Different cameras have different imaging properties.

Case 2: Different User (Shared Account)

Shared account shows that there are different users who use the same account. The security level is low when some persons use the same account.

Definition 4. Communication Information

Case 1: Same IP (wifi, PC…etc)

IP addresses are created and managed by IANA (Internet Assigned Numbers Authority). There are regional registries which are given ‘super blocks’. These regional units then allocate smaller blocks to ISP’s. If the same user logins this system by using the same IP, the security level is low.

Case 2: Different IP (wifi, PC, and etc)

Different IP One that is not static and could change at any time. This type is issued to you from a pool of addresses allocated by your ISP or DHCP Server. If the case is different, then the IP is

표 3. 사용자 IP 알고리즘

Table 3. User IP Algorithm.

```
   Algorithm : User IP
    WifiManager wifiManager=
     (WifiManager) getSystemService(WIFI_SERVICE);
    WifiInfo wifiInfo= wifiManager.getConnectionInfo();
    int ipAddress = wifiInfo.getIpAddress();
    Socket socket = new Socket("", 80);
    Log.i ("", socket.getLocalAddress().toString());
    class ip {
  String getLocalIpAddress() {
    for (Enumeration<NetworkInterface> en =
         NetworkInterface.getNetworkInterfaces();
         en.hasMoreElements();)
            NetworkInterface intf = en.nextElement();
    for (Enumeration<InetAddress> enumIpAddr =
         intf.getInetAddresses();
         enumIpAddr.hasMoreElements(); )
        InetAddress inetAddr =
        enumIpAddr.nextElement();
   if (!inetAddr.isLoopbackAddress()) then
      return inetAddr.getHostAddress().toString();
   return null;
  }
    }
```

different.

Table 3 shows user IP algorithm.

(2) CAAC Context Security Levels

We classify the user authentication into nine security levels, as shown in Table 4.

표 4. 보안 등급

Table 4. Security Levels.

Security Levels	Description
Level1	disabled, or not none
Level2	Little or no confidence (Same IP
Level3	Some confidence in channel security Same IP, Resource Rank4)
Level4	High confidence (Same IP, Same Module, Resource Rank3
Level5	Very high confidence (Same IP, Same Module, Resource Rank4)
Level6	(Same IP, Same Module, Same User, Resource Rank1)
Level7	(Same IP, Same Module, Same User, Resource Rank2)
Level8	(Same IP, Same Module, Same User, Resource Rank3)
Level9	(Same IP, Same Module, Same User, Resource Rank4)

Note: Resource Rank is the level of context resource, which use fuzzy *K*-means algorithm.

(3) Algorithm Description

Table 5 shows overall algorithm for determining security level regarding on diverse context information.

표 5. 보안 레벨 알고리즘

Table 5. Security Level Algorithm.

```
   Algorithm : Context Security Level()
   SecurityLevel()
  // SecurityLevel:Determiningsecuritylevel
   ContextComputing();
     get score[i];
     if score[i] !=null then
        for inti=0to3;    j=0 to length;
  sl[j] = score[i];
  sort(sl);//sort is a common API;
     return sl[j]; //slissecuritylevel
     if score[i] = null then
       return exception;
    end;
   ContextComputing()
  /** Determine context computing    function
      due to users' scores;
      context computing: according to the context
      percentage in context,
        the context percentage is:20%,40%,30%,10%;
   */
   SearchClassFirst();
   get a,b,c,d;
   if  ContextInformationGet()    then
  get percentage;
   /**U[a] is the data of user-a, as following;
     Every user has four data, Ua[i] means the four
  data. Ua[0] is the first data, Ua[1]is the second
  data, and so on.
   */
   for (i=0; i ≤ percentage.length; i++){
  score[0] = ΣUa[i] * percentage[i];
  score[1] = ΣUb[i] * percentage[i];
  score[2] = ΣUc[i] * percentage[i];
  score[3] = ΣUd[i] * percentage[i];
  return score[0], score[1], score[2], score[3];
   }
   SearchClassFirst()
   /**Determine the fuzzy class    function
  with the user according to    fuzzy algorithm,
  there are four classes.
            Ua, Ub, Uc, and Ud are classification
         dependent variables, where Ua, Ub, Uc, and
         Ud are the first user of classification(1) to
         classification(4), respectively;
   */
   for int j = 1 to array.length {
```

```
 String name [ ] = * readline(j).split(",");
 return string [name];
 }
  sort(name); //sort isanAPI;
  if name[0] = max(name[j]) then a =j ;
  If name[1] = max(name[j]) then b =j ;
  If name[2] = max(name[j]) then c =j ;
  If name[3] = max(name[j]) then d =j ;
  Return a, b, c, d;
```

Algorithm : ContextInformationGet()

```
  ContextInformationGet()
 /** Determine  context percentage function by using
   improved AHP method and case based  reasoning
   */
  search the user's context information and
  decide context-1,context-2,context-3, context-4;
  IAHP ();
  get context-1, context-2, context-3, context-4;
  for int i=1 to length {
     for int j=1 to length {
       percentage[i]= context[j]/100;
       return percentage;
   }
  }
```

Table 5 shows the whole algorithm of CAAC, which consists of five functions. The variables of the functions are as follow:

1. SearchClassFirst() function variables $j=(U_a, U_b, U_c, U_d)$: search the first class in fuzzy data;
2. Security level $SL=(0,1,2...)$:Thelargerthenumberis,thestrongerthestrengthis.

Fig. 4 shows the data flow of CAAC

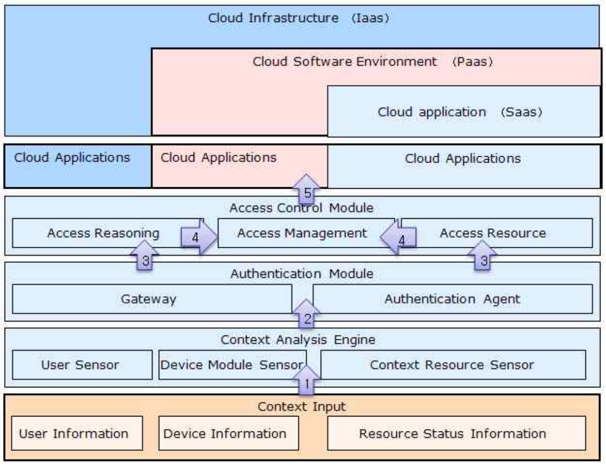


그림 4. CAAC 데이터 흐름
Fig. 4. Data flow of CAAC.

Step1: Context input module translates the input information to context engine module.

Step2: Context engine module analyzes the information and sends the analysis result to authentication module

Step3: Authentication and access control

Step4: Access reasoning module put this to access management

Step5: Access management evaluation and cloud application

Ⅳ. Result of CAAC

1. A Scenario of CAAC

Fig. 5 shows a scenario of CAAC model, where diverse contextual information may be processed by CAAC processing module which has Context Engine, Authentication Module, and Access Control Module. The result of CAAC Processing module is submitted to the Cloud Applications.

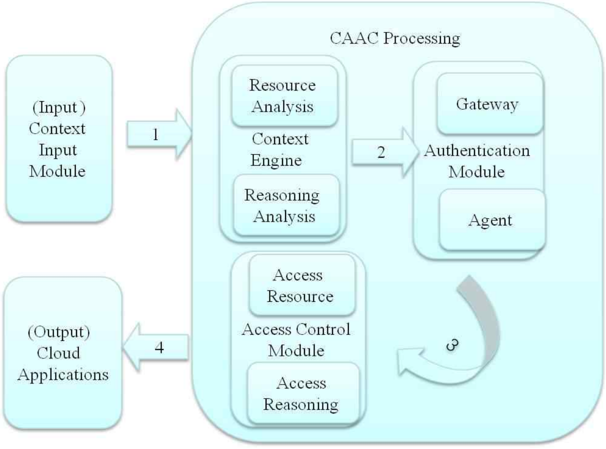


그림 5. CAAC 시나리오
Fig. 5. A Scenario of CAAC.

2. Result of the Implementation

(1) Input Data Classification

In this step, we input the user data samplings, which include location, context history, status, and login time, as shown in Table 6.

Table 7 shows the output data where these data can be separated in four classification, cluster1,

표 6. 입력 : Data 행렬
Table 6. Input : Data Matrix.

Location	Context	Status History	Login Time
16	12	13	17
26	10	15	16
16	20	30	10
…	…	…	…

표 7. 출력 : Data 행렬
Table 7. Output: Data Matrix.

Cluster1	Cluster2	Cluster3	Cluster 4
0.238142	0.661145	0.139003	0.557307
0.567673	0.052616	0.804485	0.047718
0.209392	0.149220	0.041514	0.261632
…	…	…	…

cluster2, cluster3, and cluster4.

(2) Resource Rank

The access control part includes three files, one is used to compute the first classification, the second is used to compute the context percentage, and the last one is used to compute the security level.

(20%, 40%, 30%, 10%) for instance,
Ua=U-10=17.4 C1
Ub=U-1=16 C4
Uc=U-4=18.1 C3
Ud=U-2=18.3 C2

The relation for the resource rank is as followings:
C2 > C1 > C4 > C3
The Resource Rank is L2 > L1 > L4 > L3

Table 8 shows security levels and services. Assume that one person's security level is Level 3.

(3) The Context Information

Scenario: A person named James wants to request the security services. The context information is *Same IP, Resource Rank4*, thus the security level is three.

표 8. 보안 등급 및 서비스
Table 8. Security Levels and Services.

Level	S1	S2	S3	S4	S5	S6	S7
Level 1	√						
Level 2			√	√			
Level 3	√	√			√	√	
Level 4			√	√		√	√
Level 5	√		√		√	√	√
Level 6		√	√		√	√	√
Level 7	√		√	√	√	√	√
Level 8		√	√	√	√	√	√
Level 9	√	√	√	√	√	√	√

3. Comparison of CAAC and other models

In this section we will show the context function of CAAC, which includes context table and weighting factor. And Table 9 shows comparison between CAAC model and other models.

표 9. 제안한 CAAC와 다른 모델의 비교
Table 9. Comparison between CAAC and other Model.

Features	RBAC	CAAC
Customizability	Need extended plug-in	Context analysis module can add other method
Integration	Allow non-Java applications	Allow non-Java applications
Deployment	Management role	Context analysis module
Authentication use case	Role access Role deny	LAN/Wireless/RFID
Web Service Security	Security level of server	Using fuzzy based context analysis module
Portability	Operating in a single platform	Operating in multi platforms

(1) Performance Comparison with other models

1. This model can be used in client authentication, including certain unknown context input state, hash algorithms, encryption and various properties related to security options of digital signature. By adding the context analysis engine in the middle of the model, we are able to recognize and process different types of context security, which improve better performance; in actual security system, the overall security of a class may be involved in one or a

variety of categories.

2. The output of the access control module with the cloud application, which will make good use of the security of the entire system and the cloud resources.

3. Cloud computing application separately divides cloud resource into SAAS, PAAS, and IAAS. For cloud application, different resource stands for different cloud application.

Ⅴ. Conclusion

In this paper, we proposed an authentication model security solution in authentication environment to solve the problems of the authentication vulnerability. According to authentication related theory and technology, and case based reasoning, we suggested a context-aware security model. Moreover, we embedded authentication model into authentication environment, and enhanced the security of this field. To meet most of our conditions, we considered related theory, including smart phone, Bluetooth, and RFID environment; our model will adapt the more sophisticated environments in the future, too. In this paper, we also mentioned some better method including improved AHP method in authentication environment, and some method that stick context awareness into our proposed authentication model. Additionally, we computed the context information by using improved AHP method, and hope that we can use this model in the computing of the security level for input context data.

The result of this paper is that we used context-aware algorithm to classify context information, and we expect that it can enhance the security of wireless authentication. In the future, we can further research on the fuzzy algorithm and cloud computing environment in more detail. We look forward to more secure model about authentication in the cloud computing environment.

References

[1] Saaty, T. L., *The Analytical Hierarchy Process,* McGraw Hill, New York, 1980.

[2] Intel Information Technology@, “Architecting Software as a Service for the Enterprise,” Intel Inc., October 2009.

[3] PingIdentity, http://www.pingidentity.com/.

[4] Kevin Jackon, “Secure Cloud Computing: An Architecture Ontology Approach,” *Dataline Inc.*, 2009.

[5] www.siemens.com/open.

[6] Von Altrock, Constantin. *Fuzzy logic and NeuroFuzzy applications explained*, UpperSaddleRiver, NJ:PrenticeHall PTR, 1995.

[7] McGraw-Hill Companies, Inc. http://www.mhprofessional.com/downloads/products/0072225785/0072225785_ch02.pdf.

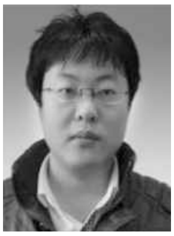
[8] Juniper, “Identity Federation in a Hybrid Cloud Computing Environment Solution Guide,” *Juniper Inc.*, http://www.boku.ac.at/mi/ahp/ahptutorial.pdf., 2009.

[9] Use case discussion group, *Cloud computing use cases*, 2009.

[10] Jiawen Chen,“fuzzy logic,” *Optimal Design Lab*, July 1999.

[11] S. Yang and M. Chung, ”Adaptive Security Management Model based on Fuzzy algorithm and MAUT in the Heterogeneous Networks,” *Journal of IEEK*, Vol. 47, No. C1-1, 2010.

저자소개



Jing Si Da(학생회원)
2007년 중국 Hustwb 대학교 학사 졸업
2011년 부경대학교 컴퓨터공학과 석사 졸업.
<주관심분야 : 인공지능, 상황인식 컴퓨팅>

정 목 동(정회원)-교신저자
1981년 경북대학교 컴퓨터공학과 학사 졸업.
1983년 서울대학교 컴퓨터공학과 석사 졸업.
1990년 서울대학교 컴퓨터공학과 박사 졸업.
1984년~1985년 금성반도체(주) 연구소 연구원
1985년~1996년 부산외국어대학교 컴퓨터공학과 교수
1999년~2000년 미국 아이오와 주립대 방문 교수
1996년~현재 부경대학교 컴퓨터공학과 교수
<주관심분야 : 컴퓨터응용보안, 인공지능, 상황인식컴퓨팅>