Journal of Information Assurance and Security 2 (2007) 250-256

Context-Aware Security Service in RFID/USN Environments using MAUT and Extended GRBAC

Kiyeal Lee[1], Seokhwan Yang[2] , Sungik Jun[3] and Mokdong Chung[1]

[1] Dept. of Computer Engineering, Pukyong National University,
599-1 Daeyeon-3Dong, Nam-Gu, Busan, 608-737, Korea
zestgame@daum.net, mdchung@pknu.ac.kr

[2] Dept. of Information Security, Pukyong National University,
599-1 Daeyeon-3Dong, Nam-Gu, Busan, 608-737, Korea
tigergal@chol.com

[3] Electronics and Telecommunications Research Institute,
138 Gajeongno, Yuseong-gu, Daejeon, 305-700, Korea
sijun@etri.re.kr

***Abstract*:** This paper proposes a context-aware security service providing multiple authentications and authorization from a Security Level which is decided dynamically in a context-aware environment. It helps developers build secure services efficiently. A security service in a dynamic environment uses Multi-Attribute Utility Theory and extended Generalized Role-Based Access Control. The system uses attribute values in GRBAC to calculate the Security Level, and extend the GRBAC. We expect this model to be widely used in providing flexible security services in a heterogeneous network.

***Keywords*:** Context-Aware Security, GRBAC, MAUT, PKI, RFID, SOM.

1. Introduction

Recently, ubiquitous technology has penetrated into almost every aspect of modern life, spreading to the furthest reaches of the world. RFID is just one such ubiquitous technology, and many people have become interested in further research in this field. RFID, however, also has a serious drawback. Individuals can tamper with such a system to obtain valuable data. As a result, data transmission in these environments is easily exposed to attacks, such as data alteration, and data forgery and disguise. To combat attacks like this, such a system should provide improved security services.

Heterogeneous networks constructed using recent network developments have diverse properties, and they may be changed dynamically according to the given environmental information. Many security models that have been proposed up to this point have been based on static security functions and policies that cannot provide adaptive enough security services for a changeable environment. We, therefore, suggest a context-aware security model which can provide security services based on users and network environmental changes using MAUT and extended GRBAC for context-aware security services.

The structure of the paper is as follows: The next section, section 2, discusses related work; Section 3 discusses a security model using MAUT and extended GRBAC; section 4 suggests a scenario and a cryptographic analysis; and section 5 concludes this paper with suggestions for future work.

2. Related work

This section describes the work of which characteristic is related to our model. Also we'll add the specific usage in or the difference from the related work as well.

2.1 EPCglobal Network

The EPCglobal network is a new standard for building RFID application. EPCglobal network consists of ALE (Application Level Event), EPCIS (EPC Information Server) and so on. The role of ALE is to provide a way to process event data, which is collected and then delivered to higher-level applications [1] – [3]. The structure and components of the EPCglobal network begin with a RFID reader that delivers identified tag data to the middleware and ALE engine. Middleware filters out various overlapping tag data, and transmits an accumulated/filtered tag data to an EPCIS or applications. We suggest a context-aware security module based on the EPCglobal Network.

2.2 Enterprise Application Framework (EAF)

The Enterprise Application Framework (EAF) is a framework that allows developers to build and use their own domain-specific RFID applications efficiently and easily. The EAF can be applied to various platforms because it is based on the standard environment, such as the EPCglobal Network, Web Services, XML, and so on. Figure 1 shows the overall structure of the EAF which was developed by our CS&AI Lab.

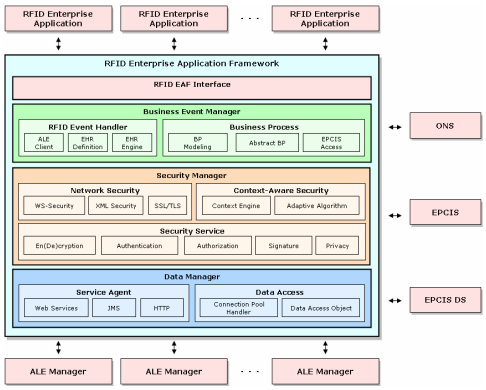

Figure 1. Enterprise Application Framework Architecture

Received November 29, 2007.

Security managers provides digital signatures, data encryption/decryption, key agreement exchanging security information, multiple authentications such as ID/PW and PKI, access control for authentication, and data protection during the transmission of data over a distributed network [4], [5].

2.3 Context-Aware Computing

Context-aware computing is a new computer paradigm that determines and utilizes certain context information, such as time and location. This paradigm can provide services which the user wants if the user's context matches context in the context-aware technology.

In Dey's definition [6], [7], context is divided into user context (such as user's preferences and age), physical context (such as location and time), computer system context (such as power on/off and devices), and non-classification context. This will be used in the development of a context-aware system according to the user's preferences.

The security model calculates the output value of GRBAC using contextual information. And MAUT adopts the output value of GRBAC as a utility value and uses to calculate the security level. The security model selects a suitable authentication module using the security level.

2.4 MAUT

MAUT (Multi-Attribute Utility Theory) [8], originally derived from economic theory, is a decision-making method using utility in making a decision based on multiple attributes. It is a systematic method that identifies and analyzes multiple variables in order to provide a common basis for decision making. As a decision making tool for predicting security levels depending on the security context, MAUT suggests how a decision maker should think systematically about identifying and structuring objectives about vexing value tradeoffs and balancing various risks.

MAUT provides a context-aware authentication method by checking a user's current environment and quantifying context information. The decision maker expresses the user's preference as a utility number through utility analysis. The utility is a relative value between 0 and 1. If we use $u(x^0)$ and $u(x^*)$ as a minimum and maximum security level utility, we can say that $u(x^0) = 0$ and $u(x^*) = 1$.

2.5 GRBAC

GRBAC is an extension of RBAC which removes subject-centric limitations, allowing the organizational power of roles for grouping environment states and objects, in addition to subjects [9]. Traditional RBAC is very useful, but it suffers from subject-centric limitations that restrict the policy designer to a subject-oriented viewpoint.

A subject role in GRBAC is analogous to the traditional RBAC role. Each subject is authorized to assume a set of subject roles. The GRBAC model allows policy designers to specify the system state through environment roles. An environment role can be based on any system state that the system can accurately collect. Object roles allow us to capture various commonalities among the objects in a system, and to use these commonalities to classify the object into roles [10], [11].

2.6 SOM (Self-Organizing Maps)

SOM is a neural network algorithm which models after mechanism of human brain that is trained using unsupervised learning [12]. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

SOM consists of input layer and competitive layer. Also learning process of SOM consists of competition, cooperation and adaptation. In competition process, SOM calculates distance among connection-strength of all neurons using input pattern, and minimum distance neuron becomes a winner. In cooperation process, only winning-neuron and neighboring neuron only learns about input vector and can modify connection-strength. In adaptation process, winning neuron and neighboring neuron update connection-strength adapting activity function which makes it more sensitive to specific input value.

Figure 2 shows network structure of SOM.

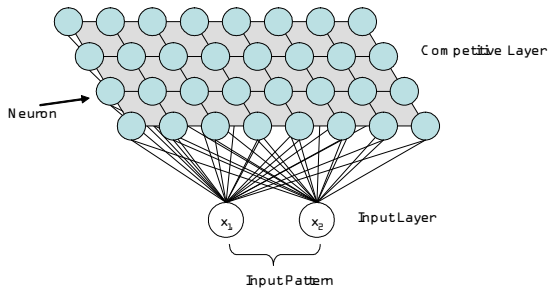

Figure 2. Network structure of SOM

Security model suggested in this paper utilizes contextual information in the RFID/USN environment, where it should handle continuously changeable input data. Since SOM uses only one feed forward flow, it has fast recognition operation, and is possible for the real-time and continuous learning. Therefore, SOM is appropriate for our context-aware security module.

Also, for more flexible security level decision making, connection-strength should be changeable for the dynamic environment. This paper applies SOM learning module to our security model, enhances its flexibility and accuracy.

3. Context-Aware Security Model

This section describes detailed contents of Context-Aware Security Model. This model suggests a security algorithm which is based on MAUT and extended GRBAC.

3.1 Overview of Context-Aware Security Model

This section explains the overall architecture of the security model proposed in this paper. The purpose of our security model is to provide diverse authentication and authorization methods according to a user's status and environment when the user wants to engage in certain transactions.

Figure 3 shows Context-Aware Security Model Architecture. This security model receives contextual data

from various terminal systems - such as a sensor, PC, network or other devices - and decides the Security Level from the data that was calculated by the Context-Aware Security Module (consisting of GRBAC and MAUT). Then this model requests a selected authentication process from the Multiple Authentication Module using the determined Security Level.

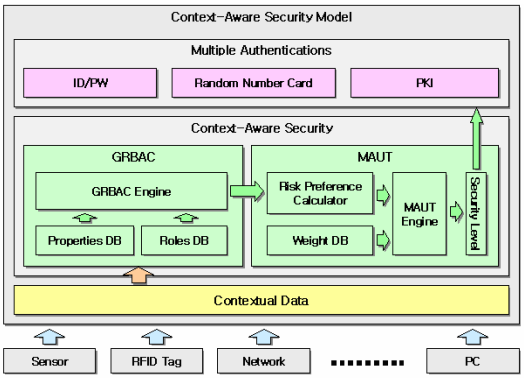

Figure 3. Context-Aware Security Model Architecture based on MAUT and Extended GRBAC

3.2 The GRBAC Algorithm for Access Control

The security model proposed in this paper utilizes GRBAC in the following way: The Subject Role is a user who demands a transaction; the Environment Role is the user's context information; the Object Role is a resource which the user wants to access; and Operation is a transaction that the user requests.

User status is defined in the Subject Role. Thus, the rest of the context information - such as access time and location - is defined in the Environment Role. The content stated above is handled in the ACS (Access Control Server) which is capable of efficient authorization.

Every service transaction of authorization is accomplished through a quadruple, T = < S, O, E, op>. This means that the Subject (S) does the operation (op) to the Object (O) in a certain Environment (E). GRBAC uses an algorithm to evaluate T.

3.3 The Extended GRBAC for Adaptive Security Level Algorithm

We will describe the extension of GRBAC mentioned in 3.2. When a user requests a service transaction, this model uses MAUT for the user to provide authentication. According to the Subject, Object, and Environment values, MAUT decides the utility value to select authentication for the current service transaction. We can express $u(x^0) = 0$ and $u(x^*) = 1$ if we set $u(x^0)$ as the lowest security grade and $u(x^*)$ the highest security grade. Since the security level is evaluated using attributes such as location, time, and resource, the total utility function is defined as shown in expression (1).

$$u(x_1, x_2, ..., x_n) = \sum_{i=0}^{n} k_i u_i(x_i), \sum_{i=0}^{n} k_i = 1 \qquad (1)$$

Where k_i is a coefficient in the subject and $u_i(x_i)$ is a utility function.

To adopt MAUT in a security model, the following requirements are necessary:

Requirement1: The Subject, Object, and Environment should have their own values.

Requirement2: A coefficient of each attribute is needed to adjust the weight of each role.

Requirement3: A risk preference is needed to decide properly a formula for the utility value according to the user's preference.

In order to satisfy these requirements, this paper extends the GRBAC mentioned in section 3.2. The process of extending GRBAC is as follows:

Algorithm 1 (Extending Object and Environment context) The GRBAC module uses information of object which the user wants to connect to. Object has its own value decided by the importance of the service in the system. In order to satisfy requirement 1, we add an attribute value to the object and environment information in GRBAC.

Algorithm 2 (Extending Subject context) In the instance of a security level determination, the administrator's location (company) and access time (working time) are more important in deciding the security level than those of the customer because the administrator should have rights to have service transactions whenever and wherever he or she wants. Thus, this paper introduces a coefficient to the attributes of Object, Environment, and Subject Roles in order to satisfy requirement 2 using MAUT theory.

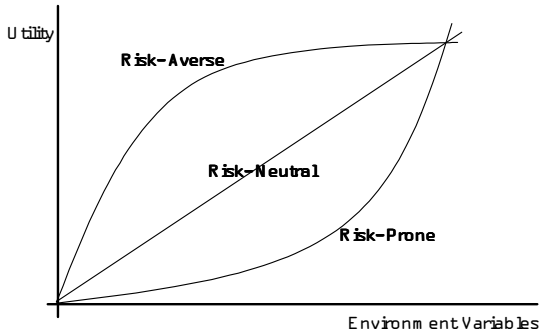

Figure 4. Three-type Risk Preference Functions

Algorithm 3 (Suggestions for user's preference) In Figure 4, risk-averse means that from the system's viewpoint the user cannot be provided with a low security authentication method while taking a high risk, and risk-prone means that the user is provided with lower authentication than others. Risk-neutral is the mixed method between both of the two like the current system. The administrator of a security model in the applied system can decide and fix the weight and risk-preference of the Subject and Object, and the attribute value of the Environment. Table 1 suggests the set context value algorithm including algorithm 1 through algorithm 3. Figure 5 shows a procedure for Security Level determination using GRBAC and MAUT. Figure 6 is an example of extending the Environment Role as

defined in 3.2. Figure 7 is an example of the extension of subject role.

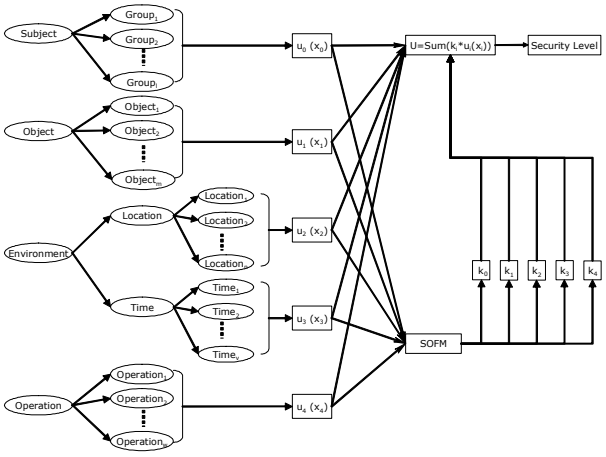

Figure 5. A procedure for Security Level determination using GRBAC and MAUT

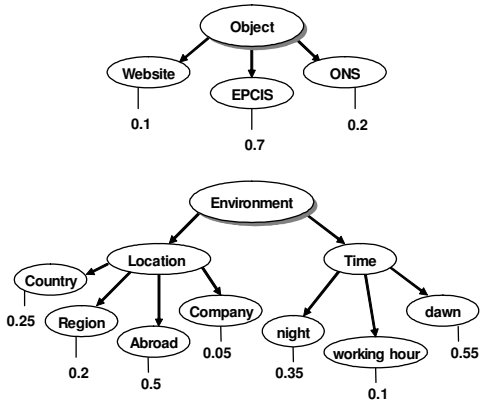

Figure 6. An Example of Extended Object and Environment Role

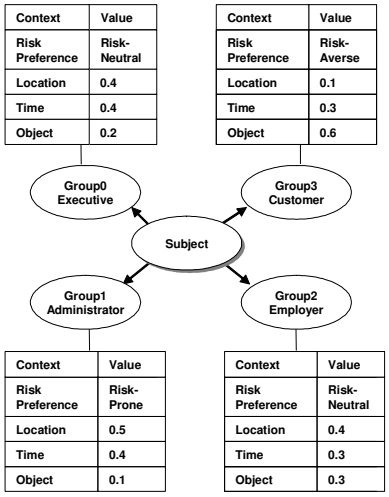

Figure 7. An Example of the Extension of Subject Role.

```
SetContextValue(subject, object, environment)
  // Algorithm 1.
    Object = GetValueOfGRBAC(object, method);
    // Algorithm 2.
    Environment.location = GetValueOfGRBAC(environment,
    location);
    Environment.time = GetValueOfGRBAC(environment, time);
    // Algorithm 3.
    Subject = GetValueOfGRBAC(subject, method);
    return ContextValue; // ContextValue consists of Subject,
                         // Object, Environment contexts
  end; // end of SetContextValue

  GetValueOfGRBAC(param, method)
   Switch (param) {

    // Algorithm 1.
    case object:
       get value from object role in GRBAC.

    // Algorithm 2.
    case environment:
       if method is location then
          get value from location in GRBAC.
       else if method is time then
          get value from time in GRBAC.

    // Algorithm 3.
    case subject:
      if method is uRiskProne then
        get value from uRiskProne role in GRBAC.
      else if method is uRiskNeutral then
        get value from uRiskNeutral role in GRBAC.
      else if method is uRiskAverse then
        get value from uRiskAverse role in GRBAC.
    default: ;

  } // end of Switch
     return value; // value is return value from each  role in GRBAC.
  end; // end of GetValueOfGRBAC
```

Table 1. The Set Context Value Algorithm.

3.4 The Adaptive Security Level Algorithm

In this section, we decide the user's security status quantitatively using a security level decision algorithm based on MAUT. We extended GRBAC to use a security method based on MAUT. Variables used in the adaptive security level algorithm consist of Subject, Object, and Environment. Values that received from GRBAC module use input of MAUT module. MAUT module computes sum of input value which multiplied with weight value (k_i) of each attributes. And then MAUT module gets security level use the computed value. Table 2 shows an adaptive security level algorithm.

The formula for deciding the security level using U, computed from the adaptive security level algorithm in the proposed model, is shown in the equation (2): SL is a quantitative value that finally decides multiple authentications, where k_i is a coefficient, $u_i(x_i)$ is a utility function and SL is the security level.

$$U = u(x_1, x_2, ..., x_n) = \sum_{i=0}^{n} k_i u_i(x_i), \sum_{i=0}^{n} k_i = 1$$
$$SL = \lceil U * 10 \rceil / 2, (SL = 0,1,...,5) \quad (2)$$

As the SL becomes higher, more and more authentication processes are required to handle the transaction.

```
SecurityLevel(Subject, Object, Environment)
```

```
// SL: Determining security level
  SL = MAUT(Subject, Object, Environment)
  return SL; end;

MAUT(Subject, Object, Environment)
  // Determine total utility function by the interaction with the
     defined GRBAC information according to MAUT
  // ki, xi, R is determined by System administrator, and will be
     changed by him or her
  // U : total utility function to determine Security Level
  // ki: set of positive scaling constants for all i in GRBAC
  // xi: domain dependent variable in in Object and Environment
     attribute value ,where ui(x°i)=0,ui(x*i)=1
  // R : Defined risk preference in Subject
  // n : The number of object and environment
     decide risk preference according to context values
     get a ki and xi in GRBAC.
     for i = 1 to n
       do ui(xi) = GetUtilFunction(xi, ki, R);
          U = U + kiui(xi)
     end
     return u(x1,x2…xi); end;

GetUtilFunction(xi, ki, R); // R is risk preference
  // Determine utility due to xi, ki, and R and is as follows
       uRiskProne   : user is risk prone for xi - convex
       uRiskNeutral : user is risk neutral for xi - linear
       uRiskAverse  : user is risk averse for xi - concave
    calculate utility of xi according to risk tendency
    if R is uRiskProne  then
       return u = (2^xi-1);
    else if R is uRiskAverse then
       return u = log2(xi +1)
    else return u = xi end;
```

Table 2. The Adaptive Security Level Algorithm.

3.5 The Learning Module

This section shows the learning module for modifying coefficient, k_i. k_i is used for computing security level value in MAUT module. Whenever the module selects inappropriate coefficient, it adjusts the value repeatedly by the help of SOM learning module. SOM module inputs from the result of GRBAC, and performs learning process to choose the reasonable coefficient.

In learning step, each neuron calculates the Euclidean distance between connection-strength and input vector, and the minimum distance neuron becomes a winner. The formula for computing of distance is defined as shown in the equation (3), where d_i is the Euclidean distance, $x_i(t)$ is the i-th input vector at time t, and $w_{ij}(t)$ is the connection-strength between the i-th input vector and the j-th output neuron.

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2 \qquad (3)$$

Only winning neuron can produce output. Also winning neuron and adjacent neighboring neurons can adjust connection-strength and learning about input vectors. This process is repeated for each input vector for a (usually large) number of cycles. The adjusting process for connection-strength is defined as shown in the equation (4), where $w_{ij}(t)$ is the connection-strength between the i-th input vector and the j-th output neuron, x_i is the i-th input vector at time t, and α is a gain term that has value between 0 and 1. α becomes smaller and smaller as time is going on.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(t) - w_{ij}(t)) \qquad (4)$$

4. A Scenario and a Cryptographic Analysis

This section describes a detailed scenario of Context-Aware Security Model. This model suggests a security algorithm which is based on MAUT and extended GRBAC.

4.1 A Scenario of Security Model

The security model proposed in this paper uses security layers from EAF. The security layer provides authentication methods using ID/PW, Random Number, PKI and authorization methods using GRBAC. Figure 8 shows a security model using EAF, MAUT, and GRBAC.

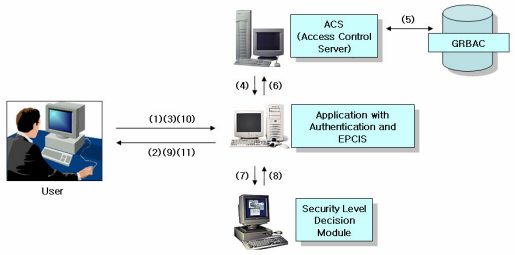

Figure 8. A Security Model using EAF, MAUT, and GRBAC

The security model proposed in this paper uses security layers from EAF. The security layer provides authentication methods using ID/PW, Random Number, PKI and authorization methods using GRBAC.

Subject	Object	Environ (Time)	Environ (Location)	Oper ation
Customer	EPCIS	10:00 AM	Region	Read

Table 3. A Customer Environment of Scenario.

(1) Customer transmits ID/PW to use application.

(2) Application authenticates user. If not, the next step will be denied.

(3) Authenticated customers ask for product information from EPCIS.

(4) Application queries ACS to check if the customer has authority to read product information. Application transmits ACS transaction information (*T=<Customer, EPCIS, (10:00 AM, Region), read>*).

If Subject Role = (all Subjects) **and** Time = (every time) **and** location = (everywhere) **then** EPCIS can be read
If Subject Role = (Administrator or Executive) **and** Time = (working time) **and** location =(company) **Then** EPCIS can be written

Table 4. An Access Policy of EPCIS for Read Operation.

(5-6) Using a GRBAC policy, ACS checks if the user is authorized to use the resource and transmits the result to Application. This policy in Table 4 means that every subject can execute read operations whenever and wherever the user needs, but executive or administrative write operation is available when the time element conforms to the working

hours. If, however, a customer requests write operations from EPCIS, the transaction is denied.

(7-8) Since the customer has the authority to read information in EPCIS, Application transmits the user's transaction information (T=<Customer, EPCIS, (10:00 AM, Region), read>) to a security level decision module to decide the security level.

(9-11) The security level is the standard value required to decide the security strength based on the user's environment. Thus, the higher the security level becomes, the more the authentication process needs to use requested resources. Figure 9 shows a sequence diagram of described scenario.

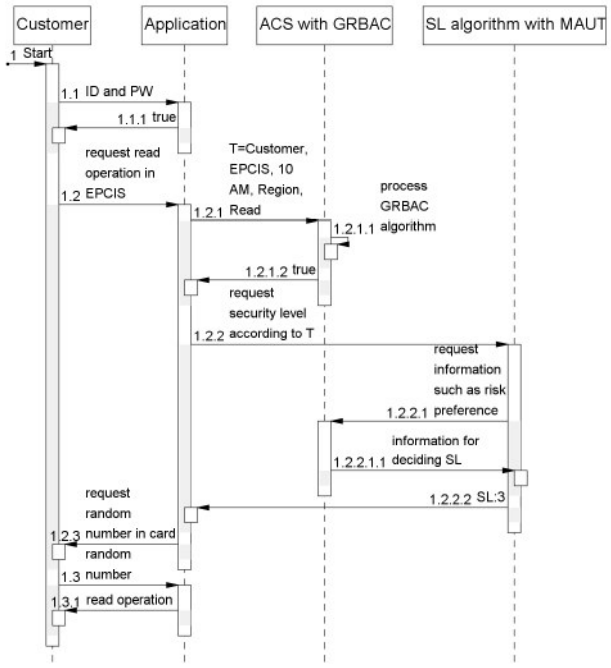

Figure 9. A sequence diagram of described scenario

4.2 The evaluation of Adaptive Security Level Algorithm

In this section, we verified the efficiency of the proposed adaptive security level algorithm. Verification goes through a comparison of the security levels of the customer, administrator and executive as decided by risk-preference under the same circumstances.

First, the security level as decided by risk-preference under the same circumstances is shown in Figure 10. The test environment is assumed to request a transaction from EPCIS with regards to sensitive data according to a time element in a domestic and foreign location. As shown in Figure 10, when the same transaction is requested in the same environment, the security level is different according to the risk-preference as defined in MAUT. When a system administrator deals with a transaction related to EPCIS, the user receives a much lower security level than the client. Therefore, the system administrator can handle transactions with greater ease and efficiency by reducing overload while having authentication.

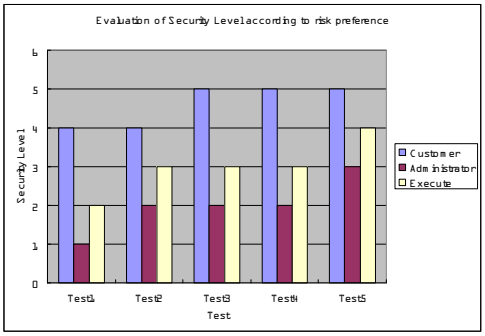

Figure 10. Testing data- load current (amperes)

Figure 11 is a result of the customer's security level according to a change of the customer's environment. We assumed the customer accessed EPCIS in situation 1, ONS in situation 2, and a website in situation 3.

From test 1 to test 12, we consider the customer's environment when the customer accesses an object. As in Figure 11, EPCIS (which manages sensitive information) has a higher security level than other items such as ONS or a website. It also shows that the customer has a different security level due to his or her environment in the same resource. Finally, we can see that the customer has a different security level from level 1 to 5 owing to the given resources and environment in a certain situation.

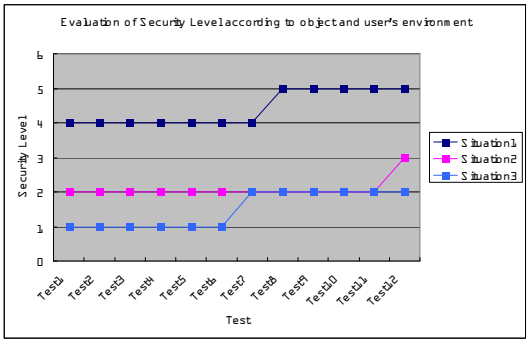

Figure 11. Result of Evaluation of SL according to User's Situation

4.3 A Comparison of Security Model

Current RFID security models do not take context-aware security into consideration. Therefore, we compared our proposed security model with typical context-aware models that have already been researched. The proposed model provides multiple authentication and authorization methods using GRBAC and MAUT in a context-aware environment. Also, we provided API through an EAF framework, thus enabling developers to build security requirement more easily.

Classification	CASA	Healthcare Application	Proposed Model
Authentication	Single authentication	Multiple authentications according to context	Multiple authentications according to context
Authorization	Authorization using	Authorization using DB	Authorization using GRBAC

	GRBAC		
Applicable pot	Mobile computing	Hospital System	Heterogeneous computing
Provides API	Not provided	Not provided	API is provided for efficient development

Table 5. The Comparison of Security Models.

5. Conclusions

This paper suggested context-aware security service in a context-aware environment. The proposed security model is based on multiple authentications, MAUT, and extended GRBAC. It might help users to use the EPCglobal network environment securely to create protected context-aware applications. We developed an adaptive security service using MAUT which helps to decide the security level according to the changes of the context-aware environment. Also, we extended GRBAC to use it in the authentication process. Therefore we could more efficiently access data defined in GRBAC. As a result, we could process more efficient authentication and access control, and could also make better RFID/USN applications by providing optimized security models for the EPCglobal Network.

In the future, we will extend adaptive security models into dynamically changeable RFID/USN environments. There seems to be a lot of administrator work overload since this model grants the work of defining roles in GRBAC to the system administrator. In order to reduce this overload, we should develop tools or technology for role definition and GRBAC extension.

Acknowledgment

This work was supported by the IT R&D program of MIC/IITA. [2006-S-041-02, "Development of a common security core module for supporting secure and trusted service in the next generation mobile terminals"]

References

[1] Auto-ID Center. EPC Information Service, White Paper, 2004.

[2] EPCglobal. Object Name Service (ONS) 1.0, Working Draft Version, 2005.

[3] EPCglobal. EPC Information Service (EPCIS) Version 1.0, Specification, 2007.

[4] Ross Anderson and Roger Needham. "Robustness Principles for Public Key Protocols," *Advances in Cryptology, Crypto 1995*, LNCS 963, pp.236–247, 1995.

[5] W. Stallings. *Cryptography and Network Security (4th)*, Prentice Hall, 2005.

[6] A. K. Dey. Providing Architectural Support for Building Context-Aware Applications, Ph. D. Dissertation, Georgia Institute of Technology, 2000.

[7] Guanling Chen. “A Survey of Context-Aware Mobile Computing Research”, *Dartmouth Univ. Computer Science Technical Report TR2000-381*, 2000.

[8] R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs (2nd Edition)*, Cambridge University Press, Cambridge, 1993.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E.Youman. "Role-based access control models," *IEEE Computer*, Vol. 29, No. 2, pp. 38 - 47, February 1996.

[10] M.J.Convington, M.J Moyer, and M.Ahamad. “Generalized Role-Based Access Control for Securing Future Applications,”. In *Proc of 23rd National Information Systems Security Conference (NISSC)*, Baltimore, pp. 115-125, 2000.

[11] M.J. Moyer and M.Ahamad. “Generalized Role-Based Access Control,”. In *Proc of IEEE Int’l Conf. on Distributed Computing Systems (ICDSC2001)*, Mesa, pp. 391-398, 2001.

[12] Teuvo Kohonen, *Self-Organizing Maps (3rd)*, Springer, 2001.

Author Biographies

Kiyeal Lee received a BS degrees in Computer Multimedia Engineering from Pukyong National University in 2006 and 2008, respectively. He is currently a member of the Computer Security & Artificial Intelligent Lab at Pukyong National University. Mr. Lee has interests in security and context-aware technology.

Seokhwan Yang received a BS degrees from Dong-Seo University in 2007 and is in Pukyong National University. He is currently a member of the Computer Security & Artificial Intelligent Lab at Pukyong National University. Mr. Yang has interests in artificial intelligent and context-aware technology.

Sungik Jun received a B.S. and an M.S. degree in Computer Science and Engineering from Chungang University, in 1985 and 1987 respectively. He joined Electronics and Telecommunication Research Institute (ETRI), Daejeon, Korea, in 1987. Now, he is team manager of wireless security application research team at ETRI. His research interests are operating system, smart card architecture and security, wireless security, and platform security.

Mokdong Chung received a Ph.D. degree in Computer Engineering from Seoul National University in 1990. He was a professor at Pusan University of Foreign Studies from 1985 to 1996. And he has been a professor at Pukyong National University since 1996. His research interests are OOP technology, computer security for application, intelligent agent, and context aware computing. He is a member of IEEE, KIISE, KIPS, KIISC, and KMMS.