A Certificate-based Context Aware Access Control Model For Smart Mobile Devices In Ubiquitous Computing Environments

Davut Cavdar, Ahmet Yortanlı, Pekin Erhan Eren, Altan Koçyiğit
Middle East Technical University, Informatics Institute,
Ankara, Turkey
Emails: {dcavdar, e129848, ereren, kocyigit }@metu.edu

***Abstract*—In this paper, a context-aware access control model is provided to be used by people with mobile devices in ubiquitous computing environments. The model utilizes a certificate-based approach and aims to create an infrastructure for regulating access requests through mobile devices to resources and services in a local environment. The model also allows users from different domains access to local resources and services within the scope of agreements between domains. In addition to conceptual design of the model, a working prototype implementation is developed and successful application of the model is demonstrated. In the prototype implementation, an application running on a real smart mobile phone is developed for generating access requests; a gateway device is utilized for context management and access control in a local ubiquitous environment with real physical sensors. Sample use cases are applied on the prototype in order to demonstrate the applicability and feasibility of the model.**

Keywords-certificate; access control; smart mobile device; context awareness

I. INTRODUCTION

Following rapid developments in the technology, smart mobile devices have become smaller than personal computers. Also, device diversity has increased to cope with the needs of humans for daily life activities. Besides standard computers, mobile devices, sensors, actuators have started to be used by humans in daily life. Unlike standard devices, these ubiquitous computing devices have effective interaction capabilities with both humans and other electronic devices. The actual contribution of ubiquitous computing is that these small and smart computers are densely distributed in the environment; they work and interact in the background invisibly and without disturbing people [12].

In today's world, humans are involved in many interactions with ubiquitous devices. Like any other system, access requests to such devices should be controlled and regulated. Especially, it is important to prevent unauthorized access to resources in ubiquitous environments.

Mainly two types of authentication methodology are offered: one is static authentication, the other is dynamic authentication. Password-based authentication is the most popular authentication method for static authentication. Smart card, biologic, Universal Serial Bus (USB) token and certificate based authentications are examples of dynamic authentication. For ubiquitous computing environments, static authentication is not suitable because access evaluation results can change according to user, location, time etc. contexts. In this paper, a certificate-based context aware access control model for smart mobile devices is provided.

This paper consists of five sections. Section 2 includes background information about certificate-based access control models. Section 3 explains the proposed model, including the main components and the activity flow. A prototype implementation of the model is introduced in Section 4. Finally, Section 5 provides the conclusion.

II. RELATED WORK

Different systems and solutions use certificates in order to regulate or check authenticity during access to resources. Web sites, mail systems, mobile applications are main areas of this usage. A user authentication method using smart cards is offered as a certificate-based authentication [10]. In this method, user certificate and other private information are stored in a smart card and the system performs authentication process based on the combination of smart card information and related context. The method focuses on authentication transactions; however, access control mechanisms and process are not discussed.

Another offered solution to access control uses certificates for access control for inter-domain environments is presented by Thompson et al. [9]. In this solution, users send their certificates storing their roles in order to reach resources However, major deficiency of this solution is that it is not designed for context aware environments and context usage.

An access control method is offered for healthcare systems by Koufi and Vassilacopoulos [6]. This method is designed for context aware environments. The system validates user roles stored in user certificates and evaluates certificate data with context information. However, this model does not support giving access to other domain users for reaching resources.

An extension model of Role Based Access Control (RBAC) is suggested for access control by Chadwick et al. [3]. The model uses X.509-based certificates that store user roles and definition for accessing resources. Access rules are defined as XML-based policy rules and they are stored in Lightweight Directory Access Protocol (LDAP) [3]. The

 ISBN: 978-1-61208-353-7

model also controls certification cancellation status using certificate revocation lists. However, the model is not suitable for context aware environments; this can be considered as the main shortcoming.

For Grid environments, certificate-based access control model namely "Sygn" is also offered by Ludwig et al. [4]. It provides decentralized permission storage and management for dynamically changing resources. Although it creates on-demand permissions without central permission systems, it is hard to regulate permissions with, in the certificates, in terms of inter-domain and security approaches.

Our proposed model combines three main properties of ubiquitous computing environments. The model provides a context aware access control and smart mobile device usage and also the model works in inter-domain environments in order to allow users access to resources of other domains.

III. PROPOSED MODEL

A. *Main characteristics of the proposed Model*

There are three main components of this model, (i) first is the user processes running on a mobile device, (ii) second is the main gateway, performing duties such as certificate control, applying rules, etc., and (iii) third is local resources or services such as reaching sensors values or printer usage. The general structure of the proposed model is shown graphically in Figure 1.

The proposed model uses certificates for authorized access to resources. After connecting to the local service wirelessly, the user sends his/her certificate to the gateway by using his/her smart mobile devices. After that, the gateway gathers required information and performs actions accordingly, and finally, produces a result. This result is received by the user via his/her mobile device again.

Another important characteristic of the proposed system is that it provides a mobile usage environment to the system users. In ubiquitous environments, computers are hidden and resources/services are widely distributed. Also, people are in transition to more mobility in terms of life styles and technology trends. Therefore, people have frequent interactions with embedded computers or resources when they are mobile. In the proposed model, home or other domain users can explore local resources/services when they are in a different location and they can send access requests to gateways.

Context-awareness is another important feature of the proposed model. In order to respond to received requests correctly according to access policy rules, the system gathers context information from the environment. Since the proposed model is a context-aware system, it senses contextual information like location, mobile user id, time, resource type and it performs required actions according to this gathered contextual information.

The proposed model can perform authorization and access control requests conducted by not only different domain users, but also by other domain users. To do that, domains provide an access policy rules agreement for their own users when they are using different domains' resources/services.

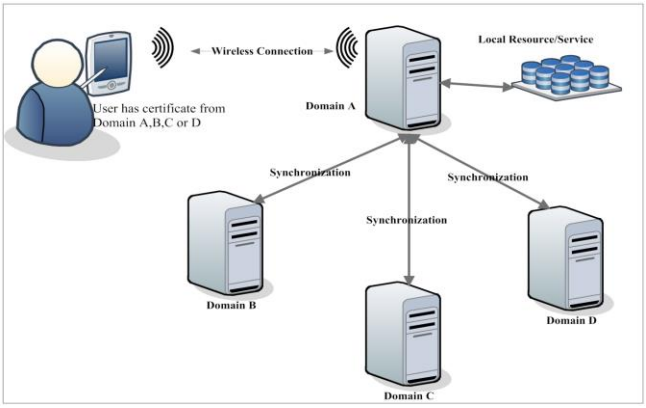

Figure 1 The general structure of the proposed model.

According to the agreement between domains, each domain sets access rules for both home and other domains' users' requests. After these agreements, the system checks other domain certificate lists in order to update access lists of other domains at each pre-defined synchronization time intervals.

There are two cases for users' domain status. When the user requests an access to a local resource/service, if s/he is home domain user, the system checks access policy rules, acquires context information, evaluates request and produces a response accordingly. If s/he belongs to a different domain, the system first checks an agreement between domains, if it is available then it checks access rules for that user, collects contextual information and finally performs an evaluation and creates a response

B. *Advantages of using certificates*

The systems need to check the identity of users for each access control by communicating with the home domain of the user. Or, at least, each domain should provide an authentication mechanism before the access decision for requests is made in order to validate user from the home domain. Such a validation mechanism requires communication between the servers of the domains and this consumes tangible amount of time and network bandwidth.

By using certificates, users carry their own authentication credentials with them so that communication between domains to validate users can be minimized. This certificate-based approach provides a faster access control compared to the approach based on authentication via the home domain. Moreover, by using certificates, users not only carry their own identity, but also carry their domain's identity with them. As a result, inter-domain service level agreement rules to access the resources can be defined in domain identity level and access control evaluations can be done based on domain identity when a user tries to access a resource with his/her certificate. That is, instead of storing users' identity in rules, storing certificate provider's identity in rules is enough in order to evaluate individual user requests.

 ISBN: 978-1-61208-353-7

C. Components of the proposed model

There are three main components of the proposed model, (i) first is user processes running on a mobile device, (ii) second is the main gateway, performing duties like certificate control, applying rules, etc., and (iii) third is local resources or services.

The gateway is the main unit of the model that accomplishes critical tasks and behaves like a bridge between the user (client) and the requested resources or services. Sub-components of the gateway are; Certificate Service, Context Engine, Decision Engine, Database Service and Management Panel.

The Certificate Service is mainly responsible for checking certificates sent by the user during access request process. After certificate information reaches the gateway, Certificate Service parses it, and checks identifier sections of the certificate. Also, the Certificate Service performs synchronization between domains. It checks domains' active certificate lists and if any change (add, delete, update) has occurred in these lists, Certificate Service updates required lists between home and other domains.

The Context Engine has mainly two duties in the proposed model. Its first task is context acquisition. Because model offers a context aware environment, context information such as location, time, group etc. about the requested resource and the user should be collected. The Context Engine collects required context information and sends them to the Decision Engine when the user demands access to the resource. Secondly, the Context Engine is responsible for managing context rules for the model. When the Decision Engine requests related rules for the defined user and resource, the Context Engine finds correct rules that will be applied for the request and sends them to the Decision Engine.

The Decision Engine is the core component of the proposed model. The user sends requests as an envelope to the Decision Engine. After receiving requests it opens envelope and defines user certificate data and resource IDs. Then, the Decision Engine demands required context information and related rules for the user from the Context Engine and sends certificate data to the Certificate Service in order to check certificate accuracy and also validity. It reaches a decision after collecting all these required data and rules.

The Database Service provides a communication infrastructure for all system modules and database. When a system module needs information stored in the database such as user group, resource ID, policy rules, it uses the Database Service to get access to the related database.

The Management Panel allows system administrators to manage system parameters by using its interface. Administrators can perform management tasks such as add or delete rules, user groups, etc., by using this panel.

D. Activity flow of the proposed model

When a user requests to reach a local resource or service via his/her smart mobile device, first s/he downloads or saves his/her certificate provided by his/her host domain into his/her mobile device, then s/he establishes a wireless connection with the resource gateway. By using the application running on the mobile device, the user selects his/her certificate and requested resource type. This type may only be one such as printer usage or more than one such as sensors providing more than one resource type like temperature, light, etc. The mobile application generates a message envelope including "Certificate Data" and "Resource Type" and sends this envelope to the gateway.

After retrieving the request envelope, the Decision Engine opens it and parses the data. Certificate data is sent to the Certificate Service for validation process. The Certificate Service first parses certificate data for default validity check, also it controls synchronized active certificate lists of other domains for certificate validity and sends the result to the Decision Engine. After that, if the certificate passes the validity check, the Decision Engine requests related contexts and rules from the Context Engine. According to the user and resource type, the Context Engine finds related rules from the database and contexts and this information are delivered to the Decision Engine. During this process, the local resource sends required data to the gateway. This data type may vary according to the designed application. If it is a file access control system, data may be up-to-date version of files, if it is a printer access control system, data may be printer current status, if it is a sensor data access control system, data may be values read by sensors.

Finally, the Decision Engine collects required information from related modules and makes an evaluation. According to the results of the evaluation, the resource/service access is allowed or denied by the system. This activity flow of the proposed model is illustrated in Figure 2 in detail.

E. Synchronization of domains' certificate lists

In the proposed model, Certificate Service performs a synchronization task in a pre-defined time period in order to make all agreed domains' active certification lists up-to-date. Each domain should be aware of certification cancellations in order to prevent access of unauthorized users into local resources. The method is based on broadcasting Certificate Cancellation List (CCL), Domains get other domains' Active Certificate List (ACL) and then each domain broadcasts its CCL in some pre-defined time intervals via its Certificate Service.

Using CCL-based synchronization method seems to be more suitable for the proposed system. However, it also has some problems; if synchronization time period is too long, this may cause unauthorized user access. An administrator of one domain may cancel one certificate; however, it may still seem to be active in the home domain due to the fact that there is time to the synchronization process.

 ISBN: 978-1-61208-353-7

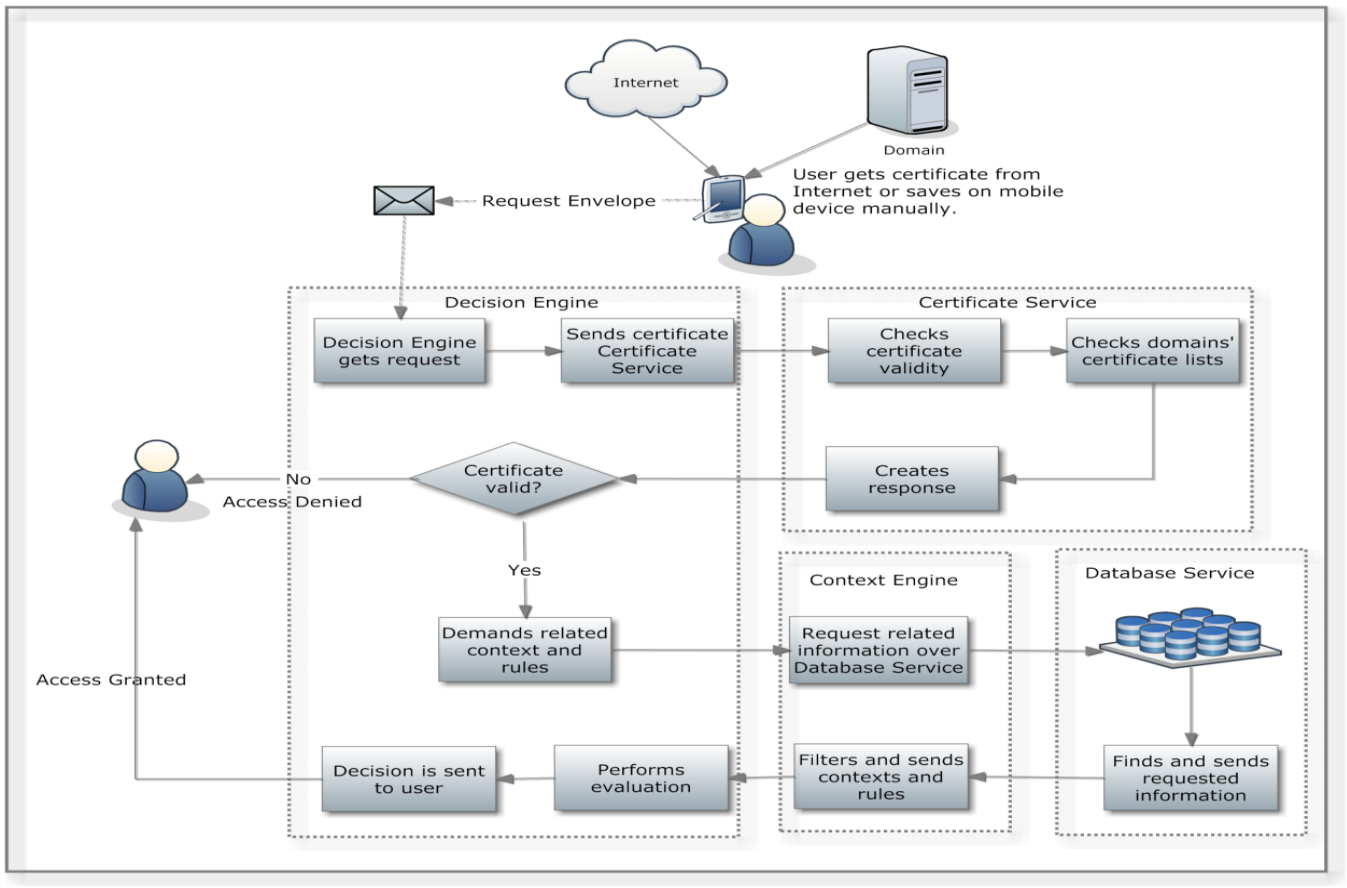

Figure 2 Activity Flow of Proposed Model

Synchronization time intervals should be set as minimum as possible according to the network communication traffic load. Another problem of the method is that when a request occurs from a different domain user, Certificate Service checks user domain's CCL, this may also cause network traffic. To overcome this problem, CCL lists of domain can be stored regularly in the home domain. These specifications and functions can be adapted according to the system environment and network conditions.

F. Management of Access Rules (AR)

The proposed model requires Access Rules (AR) in order to make decisions toward requests. The Context Engine is responsible for management, retrieving and sending required rules to requester component of the proposed model.

System Administrator defines ARs and adds them into the database. When a user requests an access, the Decision Engine asks for the related rules from the Context Engine, and then the Context Engine gets ARs using the Database Service. After getting ARs, it controls requested rules and sends them back to the Decision Engine.

ARs are transferred between the Decision Engine and the Context Engine in Extensible Markup Language (XML) format. XML is a platform and programming language-independent notation format; therefore, this usage provides flexibility for future module addition and deletion or structure change.

When a rule is requested by the Decision Engine, the Context Engine receives user's certificate/provider and resource id and then, it queries related rules with these identifiers. The Database Service finds and sends all related rules to the Context Engine. Sent rules are checked by the Context Engine and they are converted into XML structure. After that, XML based rules are sent to the Decision Engine for evaluation process. An example of Access Rule (AR):

```
<Access Rule>
<subject type = " certificate_provider "> METU
</subject>
<resource type = "service"> II/printers </ resource >
<context type = "time "> week_days </ context >
<decision > allow </ decision >
</Access Rule>
```

G. Permission evaluation method

Context Engine sends most suitable one rule to Decision Engine in order to avoid conflicts.

 ISBN: 978-1-61208-353-7

1. If there is a "deny" rule among queried rules, it has superiority over other "allow" rules for the same user and same contexts.
2. If there is no rule for requested access, Decision Engine sends "deny" response to the user.
3. If rule has time "deny" definition for current time context, Decision Engine sends "deny" response to the user.
4. If rule has time "allow" definition for current time context, however it has location "deny" definition for current location context, Decision Engine sends "deny" response to the user again.
5. If rule has "allow" definition for both time and location and if these parameters are "true" for current time and location context, Decision Engine sends "allow" response to the user and performs required operations.

IV. PROTOTYPE IMPLEMENTATION

In order to show the feasibility of the proposed model and demonstrate its applicability, a working system is developed based on the proposed model. The prototype mainly consists of an Android-based mobile application that works on a mobile device, gateway software that works on a personal computer and temperature/light sensors that work on an electronic board (microcontroller). The working logic and interactions of software modules are described in Section 3 in detail.

In the prototype, the Android-based mobile application represents mobile domain of the proposed model, J2EE-based software installed computer represents the gateway of the proposed model and temperature/light sensors represent the resource/service domain of the proposed system.

The application converts access requests into Simple Object Access Control (SOAP) envelope messages. SOAP is an XML-based messaging structure for communicating Web Services-based on Web Services Description Language (WSDL) [17].

The software in the gateway consists of five modules and these are Certificate Service, Context Engine, Decision Engine, Database Service and Data Receiver. Except the Data Receiver, other modules communicate with each other, and also with the mobile device using web services based on WSDL structure.

A. Sensor data retrieving

The microcontroller, together with the sensors mounted is connected to the gateway. The gateway detects it as a serial connection and gives it a serial port number such as "COMx". This port number is defined in the Data Receiver module of the gateway software. The Data Receiver module starts to listen to this port and after data flow starts from sensors, it detects this data and shows them in the console. After detecting, it writes these values into an external file. These values are parsed and interpreted as two different sensor values when an authorized user wants to get these data.

B. Activity flow of the prototype

Sensor data retrieving continues regularly as long as sensors work and sense the environment, therefore, this process is independent from user activity flow. Temperature sensor produces real environment temperature and light sensor measures light and gives a value between 0 and 1024 as a result of light level of environment.

The application starts with login page. The credentials on the login page are not used for authorization; they are only for program usage and can be obtained from domain administrators. After logging in successfully, the user is forwarded to the main screen of the application. In this screen, the user performs certificate transactions using two buttons. The first button "Select Certificate" forwards the user to his/her host domain to get a certificate.

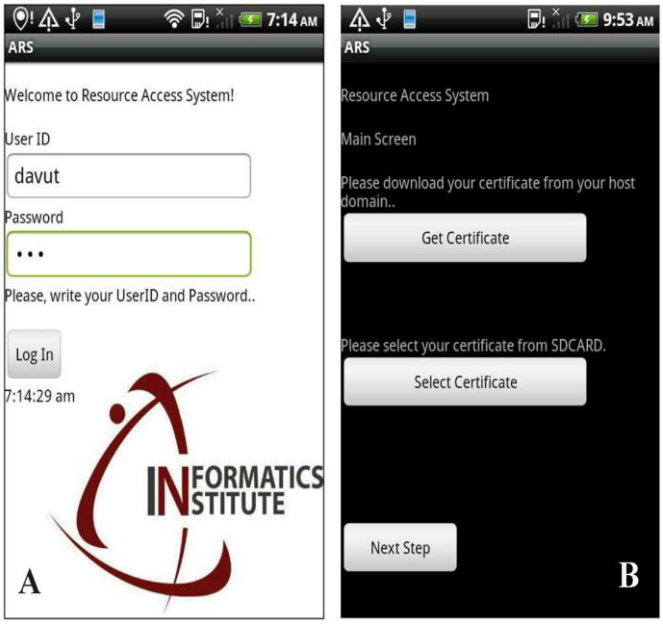

Figure 3 Login (A) and Certificate Selection (B) Interfaces of Mobile Application

The second button "Select Certificate" allows the user to select his/her certificate from the storage of the mobile device. After selecting a certificate, the application shows the certificate content for 3 seconds. If it is not possible, it gives an "unable to read file" warning. The certificate content disappears and application creates a text notification about selected certificate and its path on the SD Card. These processes are illustrated in Figure 2 (B).

The user proceeds to the final step by clicking "Next Step" and in the final interface: the application shows available resource types. According to the gateway that is connected to, the application shows which resource and resource types are available.

In the prototype implementation, available resources are sensors in the Wireless Lab of the METU Informatics Institute. This information is shown on the screen and the user selects temperature or light from drop down menu as the resource type. This selection is illustrated in Figure 3(A). With the resource type selection, the application brings together certificate data and resource type and creates SOAP access request envelope. This envelope is sent to the gateway using SOAP mechanism via the wireless connection.

 ISBN: 978-1-61208-353-7

The Decision Engine of the gateway software first receives the envelope and opens it. After opening the envelope, data is parsed, and certificate data and resource type are separated. Certificate data is sent to the Certificate Service for validity check, if certificate can pass this control, then the Decision Engine demands required context data and rules from the Context Engine. After all required data are collected, the Decision Engine performs an evaluation and makes a decision. If the user is authorized to reach temperature or light sensor data, the gateway sends related data directly to the mobile device instead of sending "allow" information only. If the user is not authorized after the evaluation process, the gateway sends "deny" response to the user's mobile device.

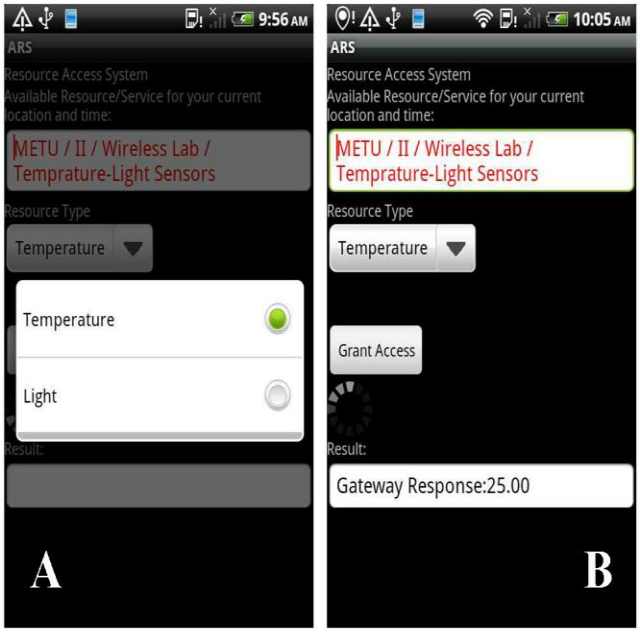

Figure 4 Resource Selection (A) and Temperature Sensor Data Response (B) Interfaces of Mobile Application

The demonstration of temperature sensor data is illustrated in Figure 3(B); also, light sensor data and access denial demonstration are illustrated in Figure 4.

C. Use cases of prototype

Different cases about trying to reach resource sensor data according to related rules and context will be analyzed. Sensors are located in the Informatics Institute (II) at Middle East Technical University (METU) and users of METU or member of user groups of METU_II and METU_CENG (Department of Computer Engineering) have different rules and privileges. Also, it is assumed that METU and Bogazici University (BOUN) have inter-domain resource usage agreement between them and users of BOUN have access to reach sensor data according to defined rules. CCL list synchronization is performed every 10 seconds. Ten different Access Rules (AR) are defined, as indicated in Table 1.

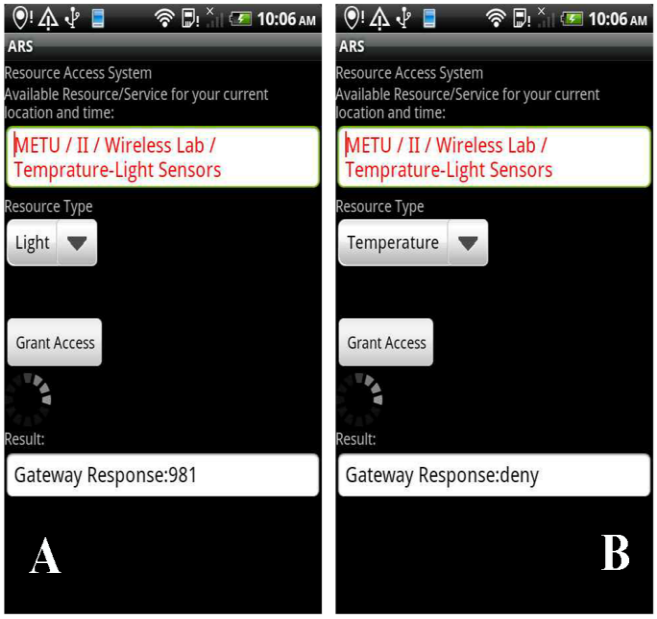

Figure 5 Light Sensor Data (A) and Access Denial (B) Interfaces of Mobile Application

TABLE 1: ACCESS RULES FOR USAGE CASES

Context	User or User Group	Resource	Response
Everytime	davut	all	allow
Everytime	serhat	light	allow
Weekend	METU_II	temp	allow
Monday	BOUN	temp	deny
Evening	serhat	temp	allow
Weekend	BOUN	all	deny
FallTerm	BOUN	light	allow
Evening	METU_CENG	light	deny
FallTerm	METU_CENG	all	allow
FallTerm	BOUN	temp	allow

D. Sample Cases

Case 1: METU domain user "serhat" wants to reach light sensor data with the following time context.
Time of request: 22.08.2011-20:30:00 (Evening, Monday)
Result: The system allows user "serhat" to access light data, because related user has two following Access Rules and first rule allows "serhat" to access light sensor data all time.

Context	User	Resource	Response
Everytime	serhat	light	allow
Evening	serhat	temp	allow

Case 2: METU_II user "ahmet" wants to reach temperature sensor data with the following time context.
Time of request: 22.08.2011-19:30:00 (Evening, Weekday)
Result: The system does not allow user "ahmet" to access light data and returns "deny" response to user , because

related user has one following Access Rule, however, related rule indicates requests with the "Weekend" time context. The time of request is not in the "Weekend" range; therefore, Context Engine does not send any rule to Decision Engine and access is not granted to the user.

Context	User Group	Resource	Response
Weekend	METU_II	temp	allow

These two and some other different usage cases are applied on the prototype implementation and they work correctly according to related rules and contexts. If the user is allowed access, the mobile application presents temperature sensor data as in Figure 3 (B), light sensor data as in Figure 4 (A) and if user access is denied, the mobile application gives "deny" response as in Figure 4 (B).

V. CONCLUSION

In this study, a certificate-based context-aware access control model using smart mobile devices for ubiquitous computing environments was presented. Using smart mobile devices for access requests and reaching resources is the major contribution of this study. In the ubiquitous computing environments, resources are distributed in the environment and in order to reach resources and use services effectively, mobile devices need to be used.

The proposed model combines three main properties of ubiquitous computing environments. The model provides a context-aware access control and smart mobile device usage and also provides inter-domain synchronization process for active certificates lists.

REFERENCES

[1] G. D. Abowd and E. D. Mynatt. "Charting Past, Present, and Future Research in Ubiquitous Computing". ACM Transactions on Computer-Human Interaction,vol. 7, no. 1, 2000, pp. 29–58,

[2] D. G. Abowd and A. K. Dey. "Towards a better understanding of context and context-awareness" 1st international symposium on Handheld and Ubiquitous Computing, London, 1999, pp. 304-307

[3] D. W. Chadwick, A. Otenko, and E. Ball. "Role-Based Access Control With X.509 Attribute Certificates" Ieee Internet Computing March- April 2003, pp. 62-69

[4] S. Ludwig, J. Pierson, and L. Brunie, "Sygn: A certificate based access control in grid environments." Tech. Report RR-LIRIS-2005-011, Laboratoire d'InfoRmatique en Images et Systmes d'information (LIRIS), 2005.

[5] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, and P. Debaty. "People, Places, Things:Web Presence for the Real World" ACM MONET (Mobile Networks & Applications Journal), 2002, pp. 365-376

[6] V. Koufi and G. Vassilacopoulos, "Context-Aware Access Control for Pervasive Access to Process-Based Healthcare Systems," eHealth Beyond the Horizon IOS Press, 2008, pp. 679-684

[7] Y. Lee, C. Min, Y. Ju, S. Pushp, and J. Song, "A Mobile Context Monitoring Platform for Pervasive Computing Environments," in 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011),, Daejeon, 2011, pp. 345-348

[8] M. Satyanarayanan, "Pervasive Computing:Vision and Challenges," IEEE Personal Communications August, 2001, pp. 10-17

[9] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari, "Certificate-based access control for widely distributed resources," in 8th conference on USENIX Security Symposium, CA, USA,1999, pp. 17-23.

[10] C.-D. Wang, L.-C. Feng, and Q. Wang, "Zero-knowledge-based user" in International Conference on Multimedia and Ubiquitous Engineering, Washington, DC, 2007, pp. 874 – 879.

[11] R. Want, "An introduction to ubiquitous computing, Ubiquitous Computing Fundamentals" J. Krumm, Ed. Redmond,Washington, U.S.A:CRC Press, ch.1, 2009, pp. 2-27

[12] M. Weiser, "The Computer for the 21st Century" Scientific American, vol. 265 no. 3, 1991 , pp. 94-104.

[13] M. Weiser and J. S. Brown, "The Coming Age Of Calm Technology," Copernicus, New York, NY, USA, 1997, pp. 75-85.

[14] B. Song, I.Yu, and J.Son, D. Baik, "An effective access control mechanism in home network environment based on SPKI certificates," Information Theory and Information Security (ICITIS), 2010 IEEE International Conference, 2010, pp. 592,595

[15] J. J. Haas, Y.-C. Hu, and K. P. Laberteaux, "Design and analysis of a lightweight certificate revocation mechanism for VANET" The sixth ACM international workshop on VehiculAr InterNETworking, New York, 2009, pp. 89-98.

[16] H. Kun, Y. Jing, D. Xiaoming, and W. Lu, "Distributed Access Control Model over Multi-trust Domain," Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on , vol. 2, 2012, pp. 595-598,

[17] Y. Kortesniemi and M. Särelä, "Survey of certificate usage in distributed access control" Computers & Security, vol. 44, July 2014, pp. 16-32

 ISBN: 978-1-61208-353-7