Activity-based Security Scheme for Ubiquitous Environments

Le Xuan Hung[1], Hassan J[1], Riaz A.S[1] , S.M.K. Raazi[1], Y. Weiwei[1], NT Canh[1], P.T.H. Truc[1], Sungyoung Lee[1], Heejo Lee[2], Yuseung Son[3], Miguel Fernandes[3], Miso (Hyoung-IL) Kim[4], Yonil Zhung[1]

[1]Dept. of Computer Engineering, Kyung Hee University, Korea,
[2]Dept. of Computer Science and Engineering, Korea University, Korea
[3]Dept. of Information Assurance, Institute for Graphic Interfaces, Korea
[4]R&D Center at Samsung Electronics.
{lxhung,hassan,riaz,raazi,weiwei,ntcanh,pthtruc,sylee}@oslab.khu.ac.kr, heejo@korea.ac.kr,{yssohn, mfernandes}@igi.re.kr, meeso.kim@samsung.com, yizhung@oslab.khu.ac.kr

Abstract — **Bardram recently introduced a new concept of activity-based computing as a way of thinking about supporting human activities in ubiquitous environments. In such environments where users are using a multitude of heterogeneous computing devices, the need for supporting users at the activity level becomes essential. However, without considering basic security issues, it could be rife with vulnerabilities. Security services, like authentication and access control, have to not only guarantee security, privacy, and confidentiality for ubiquitous computing resources, but also support user activities equipped with various devices. In this paper, we present an activity-based security scheme. The proposed scheme aims to enhance security services on mobile devices and facilitate user activities. We also integrate off-the-shell security services like MD5, TEA, Diffie-Hellman key agreement protocol so that it makes the scheme more robust and practically usable. The implementation and sample scenario have shown the requirement satisfactory of the scheme.**

Index Terms—**Ubiquitous computing, security, authentication, access control, activity-based computing.**

I. Introduction

ONE of the core ideas in ubiquitous computing [1] is to take the computer "away from the desktop" and have it pervasively available in the environment. To achieve this goal, Bardram recently introduced a new concept of Activity-Based Computing (ABC) as a way of thinking about supporting human activities in ubiquitous environments [2]. The author explored an activity-centered perspective for modeling an important class of pervasive computing systems. His main thesis is that the computing system must support handling human *work activities* directly; similar to how document-centered systems support handling documents directly. "Work activity" means well-defined tasks or processed that a person has to carry out as part of his/her job, often using computers as part of the activity. For example, you may ask an employee what he is doing, and he may answer that he is programming a module of team project; if you look at computer resource point of view, he is accessing to the project source code, and related technical documents, etc. On the other hand, the computing system must support various user devices. For example, during an interview, an employee may use a PDA; but later he may use a powerful computer for his programming work.

Traditional authentication and access control methods require much user attention and interaction in the form of manual logins, logouts, and resource browsing. Users have to remember passwords, authenticate themselves, look for what resources they need and browse the information. To be feasible to deploy on various ubiquitous devices, such as PDAs, authentication must be robust and lightweight. On the other hand, accessing to ubiquitous computing resources must be controlled in such a way that privacy preservation is guaranteed while user activities are flexibly supported. However, traditional security mechanisms exploit user identity/role information to determine the set of access permissions [4][6]-[8][14]. The policy specifications of these models tightly couple identity/role of users with their permissions. This coupling does not support user activities. Other works use context as a foundation to authorize access privileges [10]. However, the concept of context is general, for example location context, time context, system context, etc. It does not precisely specify user activities. As a consequence, these approaches are not appropriate to work in activity-centered environments.

This paper presents an activity-based security scheme which aims to support user activities in ubiquitous environments. Our security scheme is composed of image-feature based human authentication protocol [3] and activity-oriented access control model. Our contributions are in three folds: (1) we present significant variations of the protocol in [13], so as to enhance its security as well as to make it practical on various ubiquitous devices (2) we present a new access control scheme that specially aims to support user activities, the access control scheme flexibly handles user access to system resources according to current user activities, (3) we design and implement our integrated security scheme which incorporates authentication and access control schemes, along with off-the-shell security services such as Diffie-Hellman key exchanging protocol, MD5, and a number of symmetric cipher algorithms. The rest of the paper is organized as follows.

Section II briefly discusses about related work. Section III presents an overview of the proposed scheme. In Section IV, we focus on authentication and authorization services and highlight the novelty. We describe our implementation and a sample scenario in Section V. Finally, Section VI concludes the paper and outlines our future work.

II. Related Work

Many security schemes have been proposed so far [4]-[6]. However, one of the big limitations of these models is that they manage privileges based on individuals instead of group of individuals. This brings a high complexity and significant cost to manage in-growing large-scale systems. Therefore a new access control model based on the *role* of user (*Role-based Access Control* - RBAC) was introduced by D. Ferraiolo [7] to tackle this problem. RBAC is conceptually simple: access to computer system objects is based on a user's role in an organization. The authorizations are not assigned directly to particular users, but to roles. A role denotes a job function describing the authority and responsibility conferred on a users assigned to that role. RBAC is a basis for many descendant models afterwards.

Convington *et al.* [8] extended RBAC to a *Generalized Role-Based Access Control* model (GRBAC). GRBAC is an enhancement RBAC by introducing three different kinds of role. The *Usage Control* ($UCON_{ABC}$) model [9] encompasses traditional access control, trust management, and digital rights management (DRM) to control the access to and usage of digital information objects. $UCON_{ABC}$ enables finer-grained control over usage of digital objects than that of traditional access control policies and models. These approaches exploit user identity/role information to determine the set of access permissions. Therefore, they are inappropriate to apply in ubiquitous environments where access permissions are tightly coupled with the activities, rather than user identities or roles.

Corradi *et al.* proposes a new model of context-based access control, *Ubiquitous Computing Context-based Security Middleware* (UbiCOSM) [10]. UbiCOSM uses the context as a foundation for security policy specification and enforcement processes. Unlike RBAC, permissions are directly associated with contexts, instead of user identities/roles. UbiCOSM is similar to our approach in a sense that it avoids exploiting user/role information to determine the set of user permissions. However, it differs from ours that UbiCOSM's context is general (e.g. location, time, etc). This does not directly specify actual user activities.

III. Activity-based Security Scheme Overview

An overview of the scheme is illustrated in Fig. 1. The scheme is composed of authentication manager and authorization manager. Activity recognition manager provides activity information to the authorization manager. We also incorporate with off-the-shell integrity and confidentiality modules in order to make the scheme more usable. However, those modules are not shown up on the figure.

The scheme supports different types of user devices including mobile devices (PDA), laptops, computers, etc. Activity recognition manager (ARM) provides user activity information to authorization service by gathering raw contextual data related to user activity, producing high level context, and then reasoning the user actions. Authentication manager is lightweight to perform user authentication from any devices. Authorization manager bases on current user activity to look up corresponding access permissions. Then, the access trigger queries resources and sends to the user. In the following sections, we describe authentication and authorization services in more details.

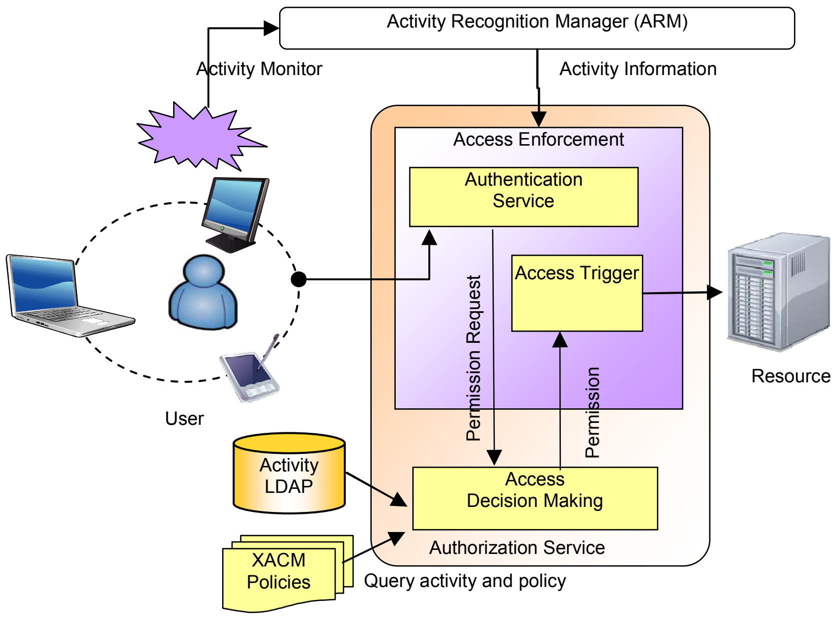


Fig 1. Activity-based Security Scheme

IV. ACTIVITY-BASED SECURITY SERVICES

A. Image-Feature based Human Identification

In the previous version [11], we have presented an image-feature based human identification to enhance security against Matsumoto threads [12]. However, the protocol cannot support mobility, as it cannot be used on mobile devices with small display units. In this version, we present significant variations, so as to make it practical on mobile devices. The protocols presented are secure under the conjectured difficulty of the problem of finding the secret feature among a given set of images, which is different from [11].

The basic concept of the protocol is as follows. The user, say Bob, and the system share a secret agreement on the image feature, for example "sport". When Bob enters his username on the dialog-box, the system bases on the username to send Bob a set of ten images. According to the images, Bob has to response either "1" or "0". Here, "1" means that the image contains *sport* feature and "0" means that it does not. For example, the correct answer for the set of images in Fig 2 is 0101001001.

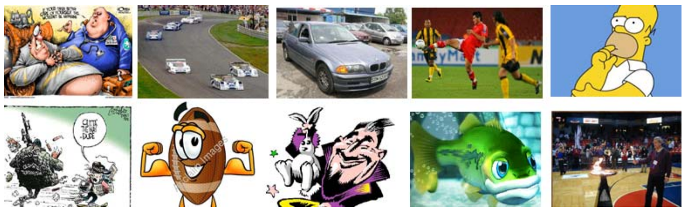


Fig 2 An example of identification images.

The main weak points of the basic protocol are as follows:

- The permutation string, if long, can be hard to remember for most of the people.
- Two users with the same secret feature might differ when answering the same image. As a result, we should allow for user error in the protocol.
- Presenting 10 images at a time is very bothersome on the user even if s/he can *scroll* all the images using a mobile device.
- In [12], whenever the system presents an image to the user, it discards it from the repository and never users it again. We can argue that if the repository of images for a given feature is large, the probability of the same image being presented in succession is very small. We can thus get rid of this impractical requirement.

In what follows, we present the enhanced protocol. We assume the set of images I and the set of features Q to be large enough so that the image is not presented again in quick succession. We denote $\mathcal{H}$ as a human user who wants to authenticate to a remote server $\mathcal{C}$. $\mathcal{H}$ is equipped with a malicious computing device. The adversary $\mathcal{A}$ enjoys all capabilities mentioned in [12].

1) *Protocol P1*

SETUP. $\mathcal{C}$ samples $q \xleftarrow{R} Q$. $\mathcal{C}$ and $\mathcal{H}$ share q as a secret.

Protocol.

- $\mathcal{C}$ initializes j $\leftarrow$ 0.
- Repeat k times.
  - $\mathcal{C}$ samples a bit $b \xleftarrow{R} \{0,1\}$. Randomly picks an image i from I such that $f(q,i)$ = b.$\mathcal{C}$ sends i to $\mathcal{H}$.
  - $\mathcal{H}$ sends the response bit a to $\mathcal{C}$, where $a \leftarrow \mathcal{H}(q,i)$
  - If $a = b$, $\mathcal{C}$ updates j $\leftarrow$ j+1
- If $j \geq s$, $\mathcal{C}$ outputs `accept`. Else $\mathcal{C}$ output `reject`.

2) *Protocol P2*

Denote by σ [m], the set of the all permutations from $\{0,1,\ldots,m\}$ to its self. Let $\sigma \in \sigma[m]$ be a generic permutation. σ [j] denotes the jth element of σ. Let `Grid` be a data structure that holds an ordered sequence of m images. The operation '+' means the append operation when applied to `Grid`.

SETUP. $\mathcal{C}$ samples $q \xleftarrow{R} Q$ and $\sigma \xleftarrow{R} \sigma[m]$. $\mathcal{C}$ and $\mathcal{H}$ share q as a secret.

PROTOCOL.

- $\mathcal{C}$ initializes j $\leftarrow$ 0.
- Repeat k' times.
  - $\mathcal{C}$ initializes `Grid`$\leftarrow$Ø.
    $\mathcal{C}$ samples $b_1 b_2 \ldots b_m \xleftarrow{R} \{0,1\}^m$.
  - For $1 \leq t \leq m$
    - $\mathcal{C}$ randomly picks an image $i_{\sigma(t)}$ from I such that $f(q, i_{\sigma(t)}) = b_t$. $\mathcal{C}$ updates `Grid` $\leftarrow$ `Grid` + $i_{\sigma(t)}$.
  - $\mathcal{C}$ sends `Grid` to $\mathcal{H}$
  - $\mathcal{H}$ initializes the answer string $a \leftarrow null$.
  - For $1 \leq t \leq m$:
    - $\mathcal{H}$ updates $a \leftarrow a \,\|\, \mathcal{H}(q, i_{\sigma(t)})$.
  - $\mathcal{H}$ sends a to $\mathcal{C}$.
  - For $1 \leq t \leq m$, if $a(t) = b(t)$, updates $j \leftarrow j$+1.
- If $j \geq s$, $\mathcal{C}$ outputs `accept`. Else $\mathcal{C}$ output `reject`.

Detailed description and security proof of the protocol can be found in [13].

B. Activity-Oriented Authorization Service

Based on characteristics of ubiquitous environments, we abstract AOAC model in three levels: user level, activity level, and privilege level, as illustrated in Fig. 3. The nature of authorization in ubiquitous environments is who is allowed to do what, for example Bob is permitted to interview a new employee. Therefore, by associating users with activity, we can easily match the authorization model into real environments. On the other hand, for user to accomplish an action, s/he needs to access to a number of resources. By connecting each activity with a set of access permissions, the proposed model highly supports user activity.

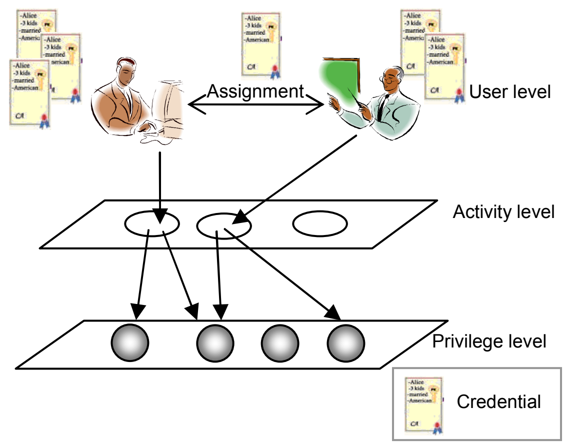


Fig.3 Abstract levels of AOAC

In AOAC, each user holds a number of credentials [15] specifying his attributes such as his role, experience, assignment. A user is authorized to perform a certain activity if his attributes are satisfied the system policies. Each activity is associated with a number of access privileges. As an example, Alice, who is a project leader, assigns Bob to develop a module *M*; since Bob is authorized to carry out 'developing module M', he has a permission to access all materials related to *M* including the source code, the technical report, the progress, etc. *Assignment* is our term to specify privilege delegation in AOAC. If Alice wants to delegate some rights to Bob, she activates an *assignment* process in which she assigns *assignment credential* to Bob. An assignment credential may carry on access rights or particular attributes (such as role, qualification, etc.) so that Bob can activate the activity.

1) Assignment

Basically, *privilege delegation* is a term to indicate that a user delegates to another user a particular privilege. In AOAC, we model privilege delegation by *assignment*, where U_1 is an *assigner* and U_2 is an *assignee*. *Assignment* is a similar term to *appointment* in [14]. It occurs when a user grants a digital credential that directly or indirectly allows another user to perform one or more activities. Credential's content may be an assignment of activities (*direct delegation*), or may be an assignment of role, qualification, etc so that the user may use to activate some activity (*indirect delegation*). Our delegation approach differs from *appointment* in several aspects. Firstly, Alice may not only delegate an access right to Bob, but Bob may also delegate this right to Carol, and so on. We call this *multi-step delegation*. Secondly, our privilege delegation may be *restricted* or *unrestricted*. It means that Alice may want to restrict how Bob can further delegate its access right. Delegation in *appointment* approach only considers how to delegate a credential to another user in order to activate one or more roles. It does not concern further delegation or restriction.

2) Privilege Revocation

Privilege revocation is very important. There are many situations that credentials should be revoked. For example, Alice delegated a credential to Bob for a particular business; if the business is accomplished, or Bob is transferred to another department, the credential should be revoked immediately. There are different reasons and different ways to revoke delegated credentials. Privilege revocation can be done in four ways [**Error! Bookmark not defined.**]:

- *By its assigner only*: only assigner can revoke the delegated credential. A credential revocation is actually a new delegation which sets negative effect on the delegated credential.
- *By anyone active in the credential*: there are some undesired situations that the assigner is not possible to revoke his delegated credential. In this case, a flexile solution is to allow anyone who is specified in the set of credential's assigners to revoke the credential on behalf of the assigner.
- *By assignee's resignation*: the assignee may resign the credential once s/he no longer wants to use it.
- *By rule-based system revocation*: revocations can be carried out by the system itself. There are different ways to be revoked: time duration on the credential is expired; constraint on the credential is violated; the task is completed; revoked at the end of the assigner session or the assignee's session.

There are two cases: *single-step revocation* and *multi-step revocation*. Single-step revocations are applied for single-step delegations. It means that Alice only delegates a credential to Bob without any further delegation from Bob. Multi-step invocations are applied for multi-step delegations. However, single-step revocations could be considered as a case of the multi-step revocation with the number of delegation step is one. There are different ways for cascading revocations. However, one of the simplest ways is based on credential identifiers (ids) to revoke them. It means that the original assigner (Alice) attaches a unique id to each credential. Whenever she wants to revoke, she just needs to indicate the credential's ids and the system will consider those credentials as invalid ones.

3) Activity Activation Rules

In order to perform an activity, the user must satisfy the conditions of the *activity activation rule*. We use Prolog-like expression to formulate our *activity activation rule* as follows:

$$ACT \vdash CON_1, CON_2, \ldots, CON_n$$

An example of *activity activation rules* is given as follows:

$$employee\_interviewing(X, Y) \vdash senior(X), personnel\_dept(X), new\_employee(Y)$$

It is notice that CON_i could be an attribute that a user holds including privileges to perform an activity, privileges to access a resource, etc. It is not restricted to only user's properties or characteristics. Attributes are encoded in X.509 certificate [15], an ITU-T standard for public key infrastructure (PKI).

4) Permission Activation Rules

Whenever a user activates an activity, corresponding permissions are automatically activated if a number of context constraints are satisfied. We define context constraints as follows:

Definition 2.1: *Context constraints* are defined as any requirement about contextual information such as time, location, etc. Context constraints play a key role to specify context-sensitive policies. In organizations, this is very important to restrict user's access and invoke user's privileges if contextual requirements are not met.

Permission activation rules are formulated as follows:

$$PERM \vdash ACT,\ CC_1,\ CC_2, \ldots, CC_m$$

i.e. if user can activate an activity *ACT* under satisfied context constraints $CC_1, CC_2, \ldots, CC_m$, then s/he is granted the corresponding permissions *PERM.*

An example of permission activation rules is:

$$read\ (X,\ employee\_profile(Y)) \vdash employee\_interviewing(X,\ Y),\ within(May,\ 2008)$$

i.e. X can access the profile of the new employee *Y* within May of 2008 during the company recruiting process if *X* is allowed to carry out interview activity.

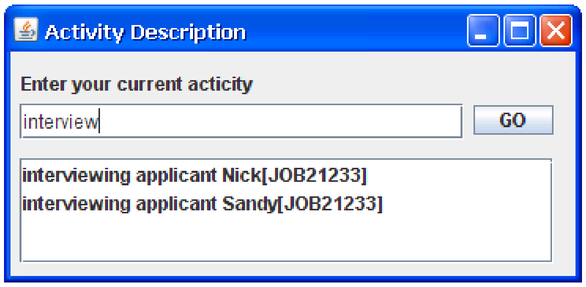
V. IMPLEMENTATION

In this section we discuss our implementation, where we use activity-based security scheme to authenticate users, capture current activity information, and make access available to user according to his/her activity. To ensure it meets our requirements, we measure the execution time of processes. We also change activity and observe how the system supports user access to resource.

The client module is deployed both on computer as well as PDA (HP iPAQ 520MHz, 64Mb RAM, and 128Mb flash ROM running Window Mobile 5.0 for Pocket PC Wi-Fi 802.11). On the server side, our security service run on PC Pentium IV 3.2GHz, 1 GB RAM, with Microsoft Window XP Pro. Authentication database is stored in My SQL server. Access policies are defined according to XACML standards [16]. User activities are stored on Lightweight Directory Access Protocol (LDAP) [17] server using Apache Directory Server (ApacheDS) [18]. For the sake of simplicity, My SQL, LDAP, and XACML are deployed on the same server. For key management, integrity, and confidentiality services, we have implemented *Diffie-Hellman Key Agreement* protocol, *Message Digest 5* (MD5), and a number of symmetric key cipher algorithms such as *Advanced Encryption Standard* (AES), *Data Encryption Standard* (DES), and *Triple DES* (3DES).

We notice that AOAC receives user activity from activity recognition manager. There are two situations of activity recognition, as discussed in [19]. First, context awareness can provide all or part of the user's intention. Second, users may explicitly state intentions.

In the former case we intended to incorporate *Activity Recognition* (AR) module from our CAMUS middleware [20]. It provides user activity information by gathering raw contextual data related to user activity, producing high level context, and then reasoning user actions. However, it is very difficult to correctly infer a satisfactory complete set of attributes to define user intent uniquely from environment monitoring and context awareness. At the implementation stage of this paper, the CAMUS's AR is not accomplished. So we decided to use the latter case.



interviewing applicant Nick[JOB21233]
interviewing applicant Sandy[JOB21233]

Fig. 4 A list of auto-searched activities is shown up as the user if typing.

In the latter case, users explicitly state intentions. The system may request activity description from a user by showing up a dialog-box "Enter your current activity". We implemented an auto-searching dialog-box, in which the system shows up a list of similar activity names as the user is typing. An example is illustrated in Fig. 4. When user is typing "interview…", the below-box shows a list of available activities so that the user can select a proper one. Though this is not a best solution, it can partly solve the problem.

Our sample scenario is as follows:

Alice, who is the project leader, assigns Bob to develop the access control module. In the meanwhile, Carol, the manager of personnel department, asks Bob to interview a new employee to support Bob in the project. As a consequence, Bob holds credentials specifying his assignment of developing access control module, and interviewing a new employee. He also holds another credential saying that he is a senior of the security team. The authorization manager inserts his new assignments into LDAP. We notice that Bob has registered an authentication id to the system and shares a secret feature 'sport'.

At eight o'clock in the morning, Bob comes to the company. He goes directly to the conference room where a new applicant, Sandy, is waiting for him. Using his PDA at hand, he authenticates himself to the system. He enters his username. Then four images are displayed on his PDA one by one. Looking at each image, Bob responses by selecting the "Yes" or "No" button (see Fig. 5). If Bob gives answer correctly on the four images, the authentication service accepts Bob as an authentic user. After that, an activity description dialog-box appears requesting to enter his current activity. As Bob selects "interviewing applicant Sandy", the authorization service verifies his permission of carrying out interviewing activity. After successful verification, it looks up LDAP and policy to get a list of matched access permissions. The access permissions include access to Sandy resume, recommendation letter, and applicant remark form (see Fig. 6). Since the screen is small to see, Bob decides to print all them out.

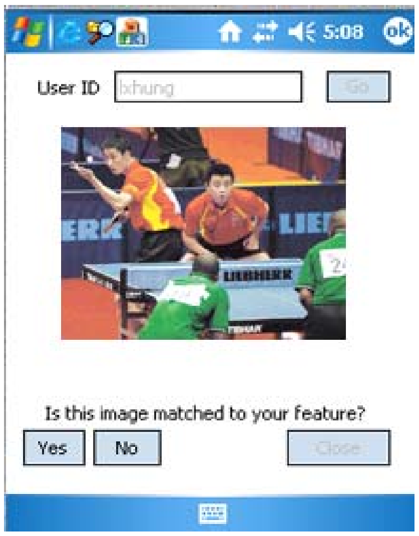
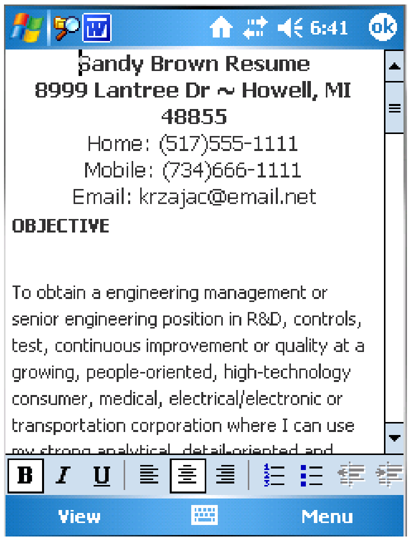


Fig. 5 Image-Feature based Human Identification



Sandy Brown Resume
8999 Lantree Dr ~ Howell, MI
48855
Home: (517)555-1111
Mobile: (734)666-1111
Email: krzajac@email.net
OBJECTIVE

To obtain a engineering management or senior engineering position in R&D, controls, test, continuous improvement or quality at a growing, people-oriented, high-technology consumer, medical, electrical/electronic or transportation corporation where I can use my strong analytical, detail-oriented and

Fig. 6 Proper resources are shown up to Bob for his interviewing

After one hour, he finishes the interview and leaves the conference room. He enters his room, turns on his computer and starts developing the module. The monitor displays the program code, project architecture, and some new project-related tasks that he has to accomplish today.

Our implementation and demonstration of the sample scenario have shown that the proposed security scheme meets the requirements. It works correctly to support user access to proper resources according to his/her current activity. Even though the scheme is deployed on PDA, it takes approximately 5.37 seconds for authentication and 0.26 seconds for authorization[1]. The reason of taking long authenticating time is because the user spent time on looking at the images and selected the right answer.

[1] The execution time is measured based on seventy repeated experiments.

VI. Conclusion and Future Work

Motivated from Bardram's work of activity-based computing, we propose an activity-based security scheme aiming to support user activities in ubiquitous environments. The scheme is also lightweight enough so that it can operate in various user devices including PDA. The security architecture comprises of authentication, access control, key management, confidentiality, and integrity. Our proposed authentication and access control are novel approaches, while the others are integrated to make a complete security solution and can be used in realistic. Authentication is based on image features to solve the drawbacks of password approach including easy guessing, pretty hard to remember, susceptible to shoulder surfing, networking snooping and key-logging. The Activity-Oriented Access Control models overcomes shortcoming of RBAC in respects that RBAC is not flexible in highly dynamic environments and privilege inheritance in RBAC is not always true in real situations. Though the implementation and sample scenario, it shows that the proposed scheme works correctly and efficiently on different types of user devices

Several issues are still ahead. One of those is how to recognize user activity based on contextual information. As our future work, we will focus more on activity recognition. Though we implemented a simple scenario, we believe that the scheme can be used for various kinds of applications such as ubiquitous life-healthcare, hospital, etc. For a future work, we will develop those applications.

Acknowledgment

This work was supported by the IT R&D program of MIC (Ministry of Information and Communication) / IITA (Institute of Information Technology Assessment). [2005-S-604-02, Realistic Virtual Engineering Technology Development].

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) " (IITA-2008-C1090-0801-0002) and by the MIC(Ministry of Information and Communication), Korea. This work was supported by the Korea Rsearch Foundation Grant Funded by the Korean Government (MOEHRD)" (KRF – 2007 – 313 - D00663).

References

[1] M. Weiser: Scientifc America. The Computer for the 21st Century. (Sept. 1991) 94-104; reprinted in IEEE Pervasive Computing. (Mar. 2002) 19-25.

[2] E. Bardram. Activity-based computing: support for mobility and collaboration in ubiquitous computing. Personal and Ubiquitous Computing, vol.9(5), pp.312-322, September 2005.

[3] Hassan Jameel, Riaz Ahmed Shaikh, Heejo Lee, and Sungyoung Lee. Human Identification through Image Evaluation using Secret Predicates. Processing of CT-RSA 2007, The Cryptographers Track at the RSA Conference 2007, San Francisco, CA, USA, LNCS 4377(2007) 67–84.

[4] Lampson, B. W., "Dynamic Protection Structures" AFIPS Conference Proc, 35, 1969, pp. 27–38

[5] Bell, D. E., and L. J. LaPadula, Secure Computer Systems: Mathematical Foundations and Model, Bedford, MA: The Mitre Corporation, 1973

[6] DoD Trusted Computer System Evaluation Criteria (TCSEC) , DoD 5200.28-STD.

[7] D. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transaction on Information and System Security, Vol. 4, No. 3, August 2001, pages 224-274

[8] M. J. Covington, M. J. Moyer, and M. Ahamad, "Generalized Role-Based Access Control for Securing Future Applications," 23rd National Information Systems Security Conference, 2000.

[9] J. Park and R. Sandhu. The UCONABC usage control model. ACM Transactions on Information and System Security (TISSEC). Volume: 7 Issue: 1 p. 128 – 174. 2004.

[10] A. Corradi, R. Montanari, and D. Tibaldi, “Context-based access control management in ubiquitous environments,” Proc. Third IEEE International Symposium on Network Computing and Applications, (NCA) pp.253–260, Aug. 2004.

[11] Hassan Jameel, Riaz Ahmed Shaikh, Heejo Lee and Sungyoung Lee: Human Identification Through Image Evaluation Using Secret Predicates. Topics in Cryptology - CT-RSA 07, Lecture Notes in Computer Science, Springer-Verlag. 4377 (2007) 67–84

[12] Matsumoto, T., Imai, H.: Human Identification through Insecure Channel. Advances in Cryptology - EUROCRYPT 91, Lecture Notes in Computer Science, Springer-Verlag. 547 (1991) 409–421

[13] Hassan Jameel, Heejo Lee and Sungyoung Lee: Using Image Attributes for Human Identification Protocols. Technical Report, CoRR abs/0704.2295, http://arxiv.org/abs/0704.2295, 2007

[14] Jean Bacon, Ken Moody, Walt Yao. “A model of OASIS role-based access control and its support for active security”. ACM Transactions on Information and System Security (TISSEC), Volume 5 Issue 4, 2002

[15] C. Adams, S. Farrell, "Internet X.509 Public Key Infrastructure: Certificate Management Protocols", RFC 2510, March 1999

[16] XAMCL and OASIS Security Services Technical Committee, "eXtendible Access Control Markup Language (XACML) committee specification 2.0," Feb 2005

[17] RFC 4511 - Lightweight Directory Access Protocol (LDAP): The Protocol

[18] Apache Diectory Server. http://directory.apache.org/

[19] Jakob E. Bardram. Activity-Based Computing - Lessons Learned and Open Issues. Workshop on Activity - From a Theoretical to a Computational Construct (ECSCW) 2005

[20] Hung Q. Ngo, Anjum Shehzad, Saad Liaquat , Maria Riaz, and Sungyoung Lee. CAMUS: A Middleware Infrastructure for Context-aware Ubiquitous Computing Systems. Technical Report, Ubiquitous Computing Lab, Kyung Hee University, Korea.