

A Secure and Flexible e-Health Access Control System with Provisions for Emergency Access Overrides and Delegation of Access Privileges

M. Fahim Ferdous Khan^a, Ken Sakamura^{ab}

^a Interfaculty Initiative in Information Studies, The University of Tokyo, Tokyo, Japan

^b YRP Ubiquitous Networking Laboratory, Tokyo, Japan

khan@sakamura-lab.org, ken@sakamura-lab.org

Abstract— Protecting electronic health records (EHR) from unauthorized access and data breaches has been a great challenge for healthcare organizations in recent times. Controlling access to EHR demands a delicate balance between security and flexibility: There are emergency cases where the default access control policy must be circumvented in order to save patients' life – and cases where management of access control rights needs to be delegated to some trusted parties. Therefore, e-Health access control systems must be robust and flexible at the same time. Conventional general-purpose access control schemes like role-based access control (RBAC) and its derivatives emphasize mainly on the robustness of the access control mechanism, and treat flexibility issues like emergency access overrides and delegation management as addenda. However, in order to comply with the care first principle of the healthcare domain, an ideal e-Health access control system should consider such flexibility issues from the ground up. Recognizing these special requirements mandated by the very nature of the healthcare profession, in this paper, we propose a secure and flexible access control system for e-Health. The user-role and object-operation mappings in our proposed system lend themselves to the RBAC model, and we implemented context verification atop this layer in order for the system to make access decision responsive to emergency incidents. For managing delegation of access control rights, we developed a secure mechanism for creation, transfer and verification of a delegation token, presentation of which to the access control system enables a delegatee to access a delegator's EHR. Every access request in our system is preceded by mandatory user authentication which we implemented using eTRON tamper-resistant cards. Security and performance analysis of the proposed system showed promising results for achieving the desired level of balance between security and flexibility required for an e-Health access control system.

Keywords— Access Control, Authentication, e-Health, eTRON, Cryptography

I. INTRODUCTION

The healthcare industry has been experiencing an IT boom. From conventional office automation to state-of-the-art digital technologies – enabled by telemedicine, wearable computing, cloud computing, IoT and big data – are pervading all sectors of healthcare enterprises, resulting in efficient operation,

greater patient outreach and improved care. However, since all these IT breakthroughs primarily focused on improving efficiency, reducing cost and incorporating value-added services, e-health has also introduced a number of crucial privacy and security issues. As more and more healthcare organizations are adopting electronic health records (EHR), instances of security breaches are increasing at an alarming rate. According to one recent survey [1], about 80% of US-based healthcare executives have reported compromise of their organizations' information technology by cyber attacks. In the first third of 2015 alone, more than 99 million healthcare records have been reported to be exposed through 93 separate attacks [2]. As EHRs contain sensitive subjective and objective information about patients, any sort of compromise thereof poses serious threats like identity theft and fraud.

One of the most important steps to ensure security of e-Health systems is to have a proper access control mechanism in place. Controlling access to EHR, however, is not straightforward as the healthcare domain presents special situations requiring exceptional access decisions. Emergency access overrides and delegation of access privileges are two examples of such exceptions. It is important to note that these are not only mere exceptions, but phenomena directly related to the very philosophy of the healthcare profession: to save life of patients at any cost. In an emergency situation, for example, if a patient suffers a sudden heart attack or shows symptoms that require immediate intervention, then she must be attended by the nearest doctor or caregiver who was not originally assigned to her (and hence access her EHR). Likewise, a deliberate departure from default management of access privileges is necessary in the case of delegation of access-control rights: Patients with Alzheimer's disease, schizophrenia, or any other mental illness – or loss of mental faculties due to old age – should be allowed by the access control system to securely delegate the management of their medical records to someone they trust. In short, e-health access control systems must serve two conflicting goals of robustness and flexibility in that they have to be sufficiently stringent to thwart any kind of unauthorized access and should also be able to seamlessly incorporate flexibility in terms of

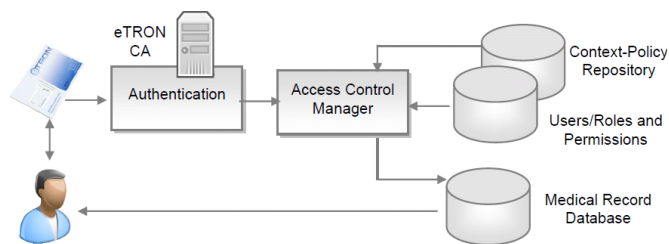


Figure 1. e-Health Access control system architecture

allowing emergency access overrides and delegation of access privileges. On this premise, in this paper, we propose a secure and flexible access control system for e-Health. We modelled the roles and permissions using role-based access control (RBAC), and implemented “emergency” as a context or attribute over this RBAC base to make the final access decision. For delegation of access privileges, our proposed system relies on a framework for creation, circulation and verification of a delegation-token, leveraging our tamper-resistant eTRON [5] and eTNet [6] security architectures.

The rest of this paper is organized as follows. Section II introduces the basic architecture of the proposed access control system. Section III and IV describes mechanisms for emergency access and delegation of access privileges. Section V briefly presents some related work, while section VI discusses the pros and cons of our proposed system.

II. E-HEALTH ACCESS CONTROL SYSTEM

As shown in Figure 1, the access control process initiates with authentication which is implemented using the eTRON architecture [5] (Section IV). Every user of the system possesses an eTRON card which has a 128-bit ID. Unique user identification is a HIPAA requirement [3], and we achieve this by using eTRON cards. The eTRON architecture implements public key cryptography where certificates are provided by the eTRON certificate authority (CA). After successful authentication, the access request will be placed to the access control manager via the session manager. The access control manager will determine the role of the authenticated user from the role and permission repository, and check the context constraints using the context-policy repository. If the requester is found to satisfy the context constraints under the role he assumes, his access request will be granted, and he will be able to exercise his access privileges congruent to the access policy. For handing delegation of access rights, we use a DAC-based approach to implement a delegation-token to be transferred to the delegate via the eTNet architecture [6].

III. EMERGENCY ACCESS OVERRIDES

A. Roles and Permissions

We used an RBAC ontology which is represented in Web Ontology Language (in RDF/XML). We used the HL7 catalogue for healthcare permissions [4] which includes a near-exhaustive list of permissions applicable to healthcare domain. However, for the sake of simplicity, we implemented only a subset of those, taking the main permissions. The ‘RBAC Reference Model Elements’ is the parent to the actual elements

or ontological concepts, namely Objects, Operations, Permissions, Role, Session, and Users – which emanate from the key RBAC terms. The ontology uses different available tags to express their intended specification. For example, class definition under RBAC_Model_Elements.

```
<owl:Class rdf:ID="Users">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="RBAC_Reference_Model_Elements"/>
  </rdfs:subClassOf>
</owl:Class>
```

The following example shows definition the ObjectProperty of `_session_role`.

```
<owl:ObjectProperty rdf:ID="_session_role">
  <rdfs:domain rdf:resource="#Session"/>
  <rdfs:range rdf:resource="#Role"/>
</owl:ObjectProperty>
```

B. Context-Policy Repository

The context-policy repository handles storage and evaluation of various context-related policies. These policies are evaluated after any access requester assumes a specific role. Some examples of how this context checking is done are given below.

Example 1: A nurse on duty can read medical data of patients that are under her responsibility only if she is requesting access during the duty hours (fine-grained “separation of duty”). A traditional RBAC would grant a nurse access to any patient’s health record even if she is not attending him.

```
user.role = Nurse & user.startDuty < time() &
user.endDuty > time(),
object.type = medical_data & object.nurseId =
user.id,
Grant{read}
```

Example 2: A nurse on duty can read the medical data of patients not under her responsibility in case of emergencies.

```
If(state = emergency)
  user.role = Nurse & user.startDuty < time() &
  user.endDuty > time(),
  object.type = medical_data,
Grant{read}
```

C. Medical Record Database

We use Oracle Database 10g Release 2 for storing the clinical data in the Medical Record Database. Oracle 10.2 is a good choice as it has some built-in functionality for supporting RBAC.

1) Role Creation: The CREATE ROLE system privilege is needed for creating roles. The command to create a role has an option to specify an activation condition that should be satisfied by the role assignee (which can be done by password or package, or externally or globally) to activate or enable a role. For example, the following command creates the role “nurse”, without requiring an activating condition.

```
CREATE ROLE nurse NOT IDENTIFIED;
```

2) User-Role Assignment: Oracle 10g allows a role to be granted to multiple users and a user to be granted multiple roles. A role can also be granted to all users in a single statement using the usergroup PUBLIC. An example of assigning the nurse role to two specific users is shown below.

GRANT nurse TO alice, carol;

3) Privilege Assignment: Privileges that can be assigned to a role in Oracle fall into two categories: system privileges and object privileges. System privileges are either commands for managing various RBAC model entities (e.g. CREATE USER, DROP USER, CREATE ROLE, DROP ANY ROLE) or commands to perform appropriate operations on different database objects (e.g., CREATE TABLE, CREATE VIEW, CREATE PROCEDURE). Object privileges, on the other hand, deal with different types of permissions on specific application objects (e.g., SELECT or UPDATE on tables).

IV. DELEGATION OF ACCESS PRIVILEGES

We realize a DAC-based delegation framework which includes three operations involving a “delegation-token”: creation, transfer and verification. Before any of these operations can proceed the communicating parties have to establish a secure session between them using a mutual authentication process as described below.

The eTRON mutual authentication is performed by a challenge-response protocol using public key cryptography. After successful authentication, a session is created to share a secure key by Diffie-Hellman algorithm. In eTRON architecture, this session is established in two phases: key generation and key confirmation. Fig. 2 shows how *A* and *B* establish a session with two publicly known Diffie-Hellman protocol system parameters, *p* and *g*. In the figure, secret key, public key, and certificate are denoted by *x*, *y* and *cert* respectively, differentiated by *A* or *B* in the subscript. *x* and *y* are related as $y = g^x \mod p$. In the first phase, *A* and *B* choose a DH key. *A* chooses a random number, R_A , and raises it to the power of *g* to get the commit value, r_A . *B* does similar things, and they both exchange the commit values and respective certificates. Later, they calculate key *K* or K' . Now, as we know DH protocol is susceptible to Man-in-the-middle (MITM) attack, we need to verify that *A* and *B* are indeed in possession of the same key, which is done in the second phase. *A* extracts the first 128-bit of *K* as w_0 , and uses it calculate a hash with *B*'s commit value and certificate. *A* then encrypts the digest with w_0 , and sends it to *B* as c_A . *B* does similar things with K' , and sends c_B . *A* then calculates V_B' as hash of w_0 , $cert_A$ and r_A . Note that V_B' and V_B differ only in the first operand inside the hash. If V_B' and V_B match, we can be sure that w_0 and w_0' , and consequently *K* and K' are same. *A* can easily verify this by decrypting c_B with its own key, w_0 , and comparing it with V_B' . In Fig. 2, $h(\cdot)$ represents SHA-1 hash function, and encryption/decryption denoted by *E/D* represents symmetric key encryption/decryption where the key is specified in the subscript of *E* or *D*.

A. Creation of Delegation-Token

The delegation-token is essentially an eTRON file (known as “electronic entity” in eTRON parlance) which includes a

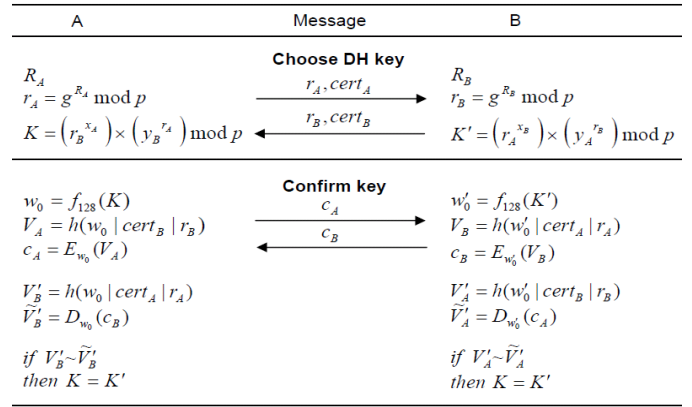


Figure 2. Session establishment protocol.

token ID, start and end dates which signify the validity of the delegation-token, and optionally name of the delegator. For creating a delegation-token the user (delegator) has to enter into a session with his eTRON card via his computer through a process of mutual authentication as described above.

In the eTRON architecture, mutual authentication has two modes: owner mode and issuer mode. By “owner” we mean owner of the card or electronic entity, and by “issuer” we mean issuer or creator of the entity (e.g., eTRON files). When an eTRON file is created, the eTRON ID of the issuer is attached to the file’s file control block. Issuer mode authentication is granted only when request to access an entity (file) comes from a party having the same eTRON ID as is written in the file control block of the entity. If the eTRON IDs match, the requesting party is given “issuer authority”. Otherwise, “other authority” is given. On the other hand, in owner mode authentication, the requesting party proves himself as the owner of the card by password or PIN, and upon successful authentication “owner authority” is granted. eTRON’s access control mechanism is based on access control lists. The file access control list in eTRON is defined by setting or resetting different bits of a 16-bit value. “Issuer” of entity (file) has all rights by default, and rights for “owner” and “others” can be set for eight access privileges, namely updating record, reading record, changing mode of a file, acquiring status of a file, deleting file, transferring file, encrypting file, and decrypting file (Table I). The file access control list is defined by the issuer of the file during the file creation process. The access control rights for the delegation-token was defined as follows. The delegate (“owner” of the card containing the delegation-token) should only be able to read the delegation-token. He should not be able to perform any update on the token, nor should he have the right to transfer the token to some other

TABLE I. FILE ACCESS CONTROL LIST IN eTRON

	Owner	Issuer	Others
Update Record	ACL0	Always Authorized	ACL1
Read Record	ACL2		ACL3
Change File Mode	ACL4		ACL5
Reference File	ACL6		ACL7
Delete File	ACL8		ACL9
Transfer File	ACL _a		ACL _b
Encrypt File	ACL _c		ACL _d
Decrypt File	ACL _e		ACL _f

persons. The verifier (authentication system of the hospital) should be able to acquire status of file to find the existence of a delegation-token, and read information inside the token to check its validity. The issuer (delegator) of the delegation-token possesses all rights by default. The following statement shows how to specify the file access control list (FACL) of the delegation-token using macros defined in eTRON library.

```

fac1 = UNL_etron_FACL_Owner_erea_rec
      | UNL_etron_FACL_Owner_eref_fil
      | UNL_etron_FACL_Other_erea_rec
      | UNL_etron_FACL_Other_eref_fil;

```

The delegation-token essentially contain the following data: a token ID, start date, end date and, optionally, name of the delegator. The start and end dates signify the validity period of the token. These data are written in the token using the *eupd_rec* eTRON API command.

B. Transfer of Delegation-Token

After the delegation-token is created, it is transferred from delegator's eTRON card to the delegate's eTRON card. For doing this, the delegator and delegate enter into a session as before. Then the transfer of delegation-token is realized using our eTNet architecture [6], which is an overlay network of eTRON devices including smartcards that enables electronic transactions to be executed securely over the Internet, without having to rely on any centrally authorized servers. After successful transfer of the token, the session is closed. The message flow in transferring a delegation-token is shown in Figure 3. Note that for the transfer operation, a special type of session, called transaction, is created between the delegator and delegate (*eopn_tra* and *ecfm_tra*). In the eTRON architecture, transaction is defined as a session with roll-back feature, which ensures that the system would return to the previous consistent state in case of any problem with connectivity or peripheral device during the transfer of a delegation-token.

C. Verification of Delegation-Token

In presence of a delegation-token in any user's eTRON card, the authentication process is modified a little. The delegate will be authenticated with his own eTRON ID, but the system will grant access to records belonging to the delegator. The system learns the delegator's eTRON ID from the delegation-token's file control block where the issuer's (delegator's) eTRON ID is written as part of the token creation process. In the verification function also, first a session is established between the delegate and verifier. After that, if the verifier finds a delegation-token (file) in the delegate's eTRON card, issuer eTRON ID is checked. For verifying the delegation token, the verification authority (hospital) gets "Other" authorization as it finds out that the issuer eTRON ID of the delegation token is different from its own eTRON ID. The verification protocol assumes that verification of delegation-token is successful if two conditions are met: the token is issued by an authorized user (registered with eTRON CA), and the token is not expired (as indicated by the validity interval written in the token file).

Note that in eTRON architecture, the two-phase session establishment requires specifying three parameters. These are

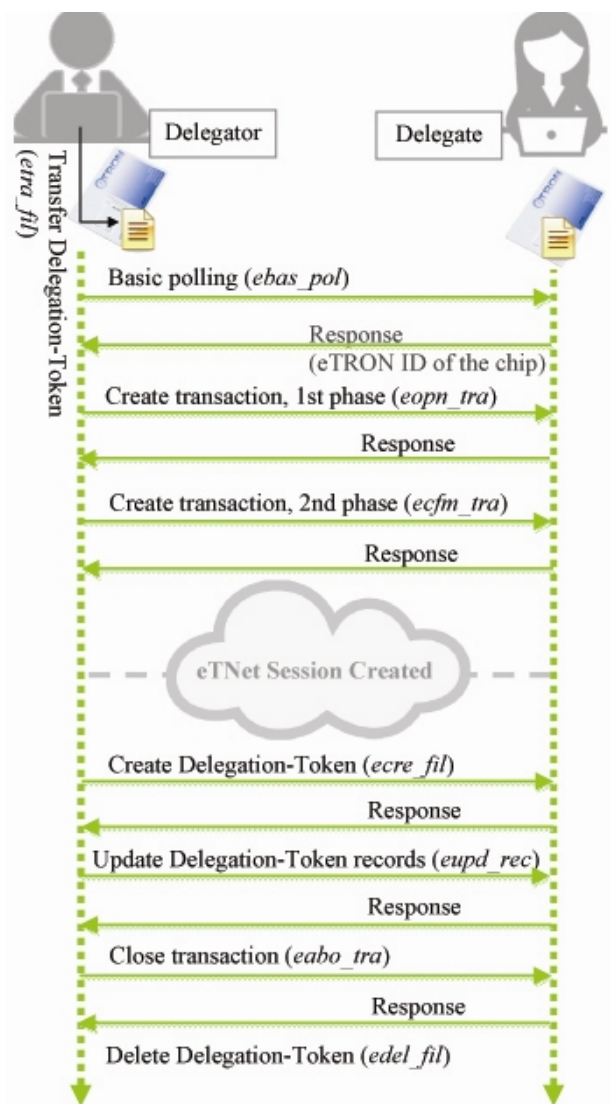


Figure 3. Transfer of Delegation-Token

the authentication mode, authentication algorithm and the crypto algorithm. The authentication mode is "issuer" for the creation and transfer operation, and "other" for the verification operation. For all three operations involving the delegation-token, we used Diffie-Hellman key sharing scheme as authentication algorithm, and Rijndael block cipher as the crypto algorithm.

We implemented the authentication protocol with eTRON cards on Microsoft Windows platform with Microsoft Windows SDK v6.1. The programming environment was Microsoft Visual C++ 2008 edition. eTRON cards are equipped with dual interfaces, i.e., they are compliant with ISO/IEC7816, and ISO/IEC 14443 standards for contact and contactless communications respectively.

V. RELATED WORK

The traditional access control models are DAC, MAC and RBAC. DAC permits the granting of access control privileges to be left to the discretion of the individual users [7]. A DAC

mechanism allows users to grant access to any of the objects under their control. MAC is a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity [8]. In its most simplistic view, RBAC – in an enterprise setting – associates users to different roles where every role is granted a proper set of permissions (i.e., operation on objects) necessary and sufficient to perform the job functions of any individual assuming that role [9].

RBAC greatly simplifies security administration, and has been adopted in many applications. Since RBAC cannot incorporate contextual attributes into access decision making, a great deal of research has been carried out in augmenting RBAC with context constraints. For example, in GRBAC model [10] context information is considered as the environmental role, which an application needs to possess in order to perform context-dependent tasks. Such a definition leads to large number of roles in an access control system since there might be potentially many environmental states that are relevant for an application. Gaia [11] defines three different role categories, corresponding to system-wide roles, active space roles, and application roles, and a mapping between them. Context-based constraints that limit a role's visibility to specific geographic areas are presented as part of the GEO-RBAC model [12]. Similarly, the GTRBAC model [13] provides mechanisms for enabling and disabling of roles based on temporal constraints.

Similarly, efforts have gone into incorporating a delegation mechanism into RBAC. Examples include RBDM0 [14], RBDM1 [15], RBDM2000 [16], PBDM [17], etc. RBDM0 enables a user in a role to delegate his role membership to another user in some other role. RBDM1 and RDM2000 support features like revocation, partial delegation, multi-step delegation, and delegation with role hierarchy. Additionally, RDM2000 introduces a rule-based language for specifying and enforcing delegation. Unlike the RBDM series, PBDM supports delegation based on both roles and permissions.

VI. DISCUSSION

The role engineering process in RBAC can be cumbersome, but once the roles are determined, it significantly reduces the complexity of security administration. It also supports review of permissions assigned to users, which aids determining risk exposure (resulting from employee system-access) by organizations which need this facility. However, RBAC has been criticized for its static nature and inflexibility, problems which can be addressed by attribute-based access control (ABAC). Thus, combining RBAC and ABAC is a very natural choice. Our approach does exactly so by implementing contextual-attribute-based access control on top of RBAC, and by treating "emergency" as a contextual attribute. To be more precise, our approach is an instance of *RBAC-A*, *role-centric* in the list of combination strategies and options for integrating attributes with RBAC (RBAC-A), as defined by Kuhn et al. [18]. This particular combination seems to be the best as contextual attribute constraints can only reduce permissions available to user (thus imposing another layer of the desired least privilege principle), and thereby the system retains the

RBAC capability to determine the maximum set of permissions that can be obtained by user [18].

Moreover, delegation – by nature – is discretionary and it is simpler to implement it using DAC as compared to approaches that try to retrofit delegation capability into RBAC. Since we are already using the eTRON architecture for authentication to initiate the access control process, using the same architecture in conjunction with eTNet architecture is a logical choice. Moreover, as the delegation-token contains the delegator's eTRON ID in its file access control block, auditing access-control delegation is possible – which other authentication-based access control mechanisms cannot support [19].

Furthermore, using the tamper-resistant eTRON architecture as an integral part of our proposed architecture offers a number of security benefits.

- eTRON uses public key cryptography for implementing a robust authentication system, which is more secure compared to smartcards that use shared key cryptography. The delegation-token is secure against duplication and alteration as operations on the electronic entities inside eTRON devices are only possible via explicit eTRON device commands. A PIN-authenticated user (or a software proxy for that user) issues commands for the eTRON device. Issuing commands is restricted based on the ACL which is defined by the issuer of entity. Even the user who owns the card cannot bypass these ACL-based restrictions. Additionally, the tamper-resistant storage of eTRON cards prevents physical attacks like microprobing.
- Man-in-the-middle (MITM) attack is not possible in the key generation phase due to checking of certificates. However, an attacker E may send her own commit value, r_E to both A and B . Obviously, doing so will not enable E to establish a pair of keys with A and B – as depicted in classical MITM scenario – as she cannot forge A or B 's private key. Nevertheless, such a scenario would impede A and B to enter into a secure communication session. Had there been no key confirmation phase, this anomaly would have been undetected.
- The peer-to-peer transfer of delegation-token via eTNet is inherently more secure as it does not require assuming trust on any intermediate central servers. The eTNet architecture has also been proved to be secure against MITM attacks [6].

In short, the proposed access control scheme possesses different advantages of RBAC (e.g., user/role management, principle of least privilege and separation of duty), and the simplicity of DAC for delegation management. Our access control mechanism also includes the pre-requisite authentication step. Moreover, our approach obeys guidelines defined by HIPAA Security and Privacy Rule [3]. For example, emergency access, network security, access authorization, principle of minimum disclosure, user consent, etc. Furthermore, using HL7 standards for defining the RBAC roles and privileges lends a degree of interoperability to our approach.

An important future work to enhance our proposed scheme is to develop a mechanism for post-fact verification of emergency accesses as such security overrides may cause potential access-control abuses, and therefore, a strong auditing mechanism has to be in place.

VII. CONCLUSION

Access control in e-Health requires both robustness and flexibility. This paper underscores the importance of these dual requirements and the balance between them. To this end, we have proposed an access control scheme capable of handling emergency access overrides and delegation of access privileges in addition to providing standard access control. The proposed e-Health access control system is self-contained in that it implements the pre-requisite authentication step, whereas most existing access control models usually assume one to be in place. The authentication step is realized using the tamper-resistant eTRON security architecture. The security features of eTRON have also been leveraged for implementing the delegation of access privileges in a secure manner. Our system treats emergency situations as contextual attributes, and models those over a RBAC framework. As the delegation of access privileges is an instance of discretionary access control in our design, the proposed system is essentially a combination of RBAC and DAC. We are of the view that such judicious combination is necessary as retrofitting any particular model to satisfy all access-control requirements of e-Health is prohibitively expensive.

ACKNOWLEDGMENT

We cordially thank the YRP Ubiquitous Computing Laboratory (www.ubin.jp) for providing the eTRON hardware under the “Secure Ubiquitous Computing Platform” project supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

REFERENCES

- [1] Health Care and Cyber Security – Increasing threats require increasing capabilities. August 2015 [Online] <http://tinyurl.com/srvy2015>
- [2] Perspectives on Cybersecurity in Healthcare. June 2015 [Online] <http://www.wedi.org/docs/test/cyber-security-primer.pdf>
- [3] HIPAA, Health Insurance Portability and Accountability Act, <http://www.hhs.gov/ocr/privacy/index.html>
- [4] HL7, Health Level Seven International, <http://www.hl7.org/>
- [5] K. Sakamura and N. Koshizuka, “The eTRON Wide-Area Distributed-System Architecture for E-Commerce,” *IEEE Micro*, Vol. 21, No. 6, 2001, pp. 7-12.
- [6] T. Yashiro, M. F. F. Khan, S. Ito, M. Bessho, S. Kobayashi, T. Usaka, N. Koshizuka and K. Sakamura, “eTNet: A Smart Card Network Architecture for Flexible Electronic Commerce Services,” 4th IFIP Int’l Conference on New Technologies, Mobility and Security, 2011, pp. 1-5.
- [7] National Computer Security Center (NCSC), “A Guide to Understanding Discretionary Access Control in Trusted System,” Report NSCD-TG-003 Version1, September 1987.
- [8] Department of Defense, “Trusted Computer System Evaluation Criteria,” DoD 5200.28-STD, National Computer Security Center, Ft. Meade, MD 20755, December, 1985.
- [9] R. S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, “Role-Based Access Control Models,” *IEEE Computer*, 29(2), pp. 38-47, 1996.

- [10] M. J. Covington, W. Long, S. Srinivasan, A. K. Dey, M. Ahamad, and G. D. Abowd, “Securing Context-Aware Applications Using Environment Roles,” In *Proc. of the Sixth ACM Symposium on Access control Models and Technologies*, 2001, pp. 10-20.
- [11] G. Sampemane, P. Naldurg, and R. H. Campbell, “Access control for Active Spaces,” In *Annual Computer Security Applications Conference*, 2002, pp. 343-352.
- [12] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, “GEO-RBAC: A Spatially Aware RBAC,” In *Proc. of the Tenth ACM Symposium on Access control Models and Technologies*, 2005, pp. 29-37.
- [13] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, “A Generalized Temporal Role-Based Access Control Model,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 1, pp.4-23, 2005.
- [14] E. Barka and R. Sandhu, “A Role-based Delegation Model and Some Extensions,” *Proceedings of the 23rd National Information Systems Security Conference*, pp.101-114, 2000.
- [15] E. Barka and R. Sandhu, “Role-Based Delegation Model/Hierarchical Roles,” *Proceedings of the 20th Annual Computer Security Applications Conference*, pp.396-404, 2004.
- [16] L. Zhang, G. Ahn and B. T. Chu, “A Rule-Based Framework for Role Based Delegation,” *Proceedings of the 6th ACM Symposium on Access Control Models and Technologies*, pp.153-162, 2001.
- [17] X. Zhang, S. Oh, and R. Sandhu, “PBDM: A Flexible Delegation Model in RBAC,” *Proceedings of the 8th ACM Symposium on Access Control Models and Technologies*, pp.149-157, 2003.
- [18] Kuhn, D. R., Coyne, E. J. and Weil, T. R., “Adding attributes to role-based access control,” *IEEE Computer*. 43, 6 (June 2010), pp. 79-81.
- [19] Karp, A. H., Haury, H. and Davis, M. S. 2009. From ABAC to ZBAC: The evolution of access control models. Technical Report HPL-2009-30, HP Labs. Available: <http://www.hpl.hp.com/techreports/2009/HPL-2009-30.pdf>



M. Fahim Ferdous Khan, PhD is an assistant professor at the Graduate School of Interdisciplinary Information Studies in the University of Tokyo. He obtained his Master’s and Doctoral degrees from the same university in 2009 and 2012 respectively. Prior to joining the University of Tokyo, he had been serving as a lecturer at the Department of Computer Science and Engineering in Islamic University of Technology, Bangladesh. His current research focus includes developing resource-aware security mechanisms for the Internet of Things and cyber-physical systems, and exploring the intersection of context-awareness and security in ubiquitous computing. He is also investigating various security, privacy and trust issues related to e-commerce, smartcard, RFID and other emerging distributed applications. He is a member of the IEEE, IEEE Computer Society, IEEE Communications Society, and the ACM.



Ken Sakamura, PhD is a professor at the Graduate School of Interdisciplinary Information Studies in the University of Tokyo, and the director of Institute of Infrastructure Application of Ubiquitous Computing there. As the founding leader of the TRON project initiated in 1984, he has designed the TRON open computer system architecture, which has been widely used in many consumer electronics appliances including mobile phones, digital cameras, FAX machines, engine control of automobiles, etc. Currently, as the chairman of the TRON Forum (www.tron.org) and the Ubiquitous ID Center (www.uidcenter.org), he has been leading cutting-edge IoT and ubicomp research. He has held the position of the director of YRP Ubiquitous Networking Laboratory since January 2002. He has been elected as fellow and golden core member of the IEEE Computer Society. He has won numerous awards, most notably, the Takeda Award in 2001, the Medal with Purple Ribbon from Japanese government in 2003, Okawa Prize in 2004, Prime Minister Award in 2005, Japan Academy Prize in 2006 and ITU150 Award in 2015.