Formal Policy based Authorization Model for Ubiquitous Enterprise Computing Environment

Supreet Kaur
PURCITM, Punjabi University,
India

Kawaljeet Singh
UCC, Punjabi University,
India

ABSTRACT
Ubiquitous computing environment demands a dynamic access control mechanism that can adapt to the changing security requirement of the computing environment. In this paper an authorization model for ubiquitous computing environment is proposed and a formal approach is adopted to design a flexible and scalable model to support intelligent authorization process in ubiquitous computing environment.

Keywords
Access Control, Authorization, Formal Methods, Security Model, Ubiquitous Computing.

1. INTRODUCTION
Ubiquitous computing environment, a concept proposed by Mark Weiser [1-2] is an emerging computing environment in which people can do anytime and anywhere computing in a fully interconnected multi domain environment. The traditional security models need to be enhanced to deal with the new security requirements. In this paper the main objective is to investigate the security issues associated with authorization service in ubiquitous computing environments and propose a well designed formal model based on the notion computational intelligence that integrate multiple parameters from different security paradigms and adapt well with computing environment.

The proposed model has a modular architecture in which security modules can be incorporated in the security policy to decide the authorization. The modular approach enables dynamic adaptation of policies for changing security requirements of target computing environment. The main security component is the security knowledgebase that implicitly provides computationally intelligent security framework at the backend for authorization model.

2. RELATED WORK
In past significant amount of research work has been done in the area of access control model and policies. The work done by Lampson [3] can be considered as foundation for formal approach towards access control technologies. The traditional access control models [4-6] were developed keeping in view the specific access control requirements of the system. These models were able to meet the protection requirement of the system through single policy framework and mechanism. With development of new computing environment, the traditional models were unable to meet multiple policy requirements through single access control mechanism. There is need for a flexible and scalable authorization model that can meet the different protection requirement of the computing system through single security mechanism. In this direction various extensions to the traditional models has been proposed in literature to address the security issues of emerging computing environment. The various studies in this context is outlined below. In research work [7], authors proposed a context centric access control model for ubiquitous and mobile computing environment by taking into account different types of metadata. Lin et al.[8] present a flexible, autonomous and non-redundancy access control model for ubiquitous computing environment which dynamically grants and adapts permissions to users based on context information including time, location and trust value. Hung et al.[9] proposed Activity-Oriented Access Control (AOAC) model, aiming to support user's activity in ubiquitous environments. Sejong [10] proposed a new access control model termed the Ubi-RBAC models based on the RBAC model and adds new components such as space, space hierarchy, and context constraints. Manachai et al.[11] proposed a spatio-temporal access that can be used by any application where the access is contingent not only on the role of the user, but also on the locations of the user and the object and the time of access. Sigrid et al.[12] in their work integrated context constraints with process-related role-based access control (RBAC) models and presented model that supports context-dependent task execution. The work proposed in the section 3, takes into the consideration the concept of policy based access control model with emphasis on the concepts of intelligent knowledgebase oriented authorization process for the development of effective authorization system for secure ubiquitous computing environment.

3. UBIQUITOUS AUTHORIZATION FRAMEWORK
The Authorization Model is used to formally represent the set of authorization policies. Formal modeling approach helps to verify and validate the security properties of the Authorization system for which it is designed. The authorization mechanism is the enforcement of the Authorization policy formally stated through Authorization model.

In Ubiquitous computing environment, the entities that pair up for interaction may be unknown to each other and system may not have any past record of entities. In such case the system relies on the knowledge base developed over a period of time about the entities and the environment. In Ubiquitous computing environment the access decision depends on application of multiple access policies under different policy domain. There is a need of a model that is able to capture

complexity of the target environment and must enforce the authorization process based on multiple factors based authorization policy. In the next section the development of formal authorization model is described. The proposed model has different components which are used to develop authorization framework for ubiquitous computing environment.

3.1 Development of Formal Ubiquitous Authorization Model (UAM)

To provide secure authorization service a formal Ubiquitous Authorization model (UAM) is proposed for the specification of authorization security property of ubiquitous computing environment. The UAM specification provides the detailed information of different components used to develop authorization process that will allow only the legitimate access of resource in the computing environment.

3.1.1 UAM Model Specification

In order to specify the Ubiquitous Authorization model a state machine based formal approach is considered to define system model as an abstract state transition system. With respect to Ubiquitous computing environment the proposed abstract state machine system compose of set of states, system entities, set of operations, transition functions, authorization evaluation function. The UAM generic specification is a 5 tuple as follow.

$$UAM = \langle U_{SS}, U_{AR}, U_{AuthStateP}, ST_{UAM}, U_{IS|s_0} \rangle$$

where

U_{SS} : Set of Ubiquitous System States

U_{AR} : Set of Ubiquitous Access Request Operation

$U_{AuthStateP}$: Authorized State Permission Set

ST_{UAM} : System Transition function

$U_{IS|s_0}$: Initial System State.

Fig 1: The Generic specification of Formal Ubiquitous Authorization Security Model.

Ubiquitous System State: A state $s \in U_{SS}$ represents the current state of the Ubiquitous computing system and holds the necessary information required to make access control decision.

Access Request: The access request $ar \in U_{AR}$ is a user request to access a particular resource in Ubiquitous Computing system.

Authorization State Permission: The authorization policy equation $p \in U_{ASP}$ is a multi attribute policy equation that decides whether the user access request is allowed or not allowed in a given state.

State Transition Function: The state transition function $tf(MOp, s, s') \in ST_{UAM}$ represent set of operation or actions with preconditions .The application of transition function results in state change that can be represented as $tf(Mop, s, s') \rightarrow (s, s')$ where $s, s' \in U_{SS}$.

The state of the system consists of subjects, objects and set of current accesses allowed after evaluation of access request by authorization evaluation equations. The UAM system constrains accesses by considering multiple parameters profile associated with each entity active in ubiquitous computing environment. The security information of UAM system consists of two components: the security parameter SP_{UAM} and the security function SF_{UAM}. The security parameters represent the security relevant characteristics used by the authorization mechanism and depend on the target computing environment. All the relevant security parameters need to be captured and modeled as part of state machine to build reliable authorization mechanism. The security functions SF_{UAM} are used to bind the security information with the system entities dynamically. The security information with respect to system entities may change with the change of the state. Let $s \in U_{SS}$ be a state of the system for the ubiquitous computing environment. The state is composed of set of access request $ar \in U_{AR}$ initiated by subject on behalf of the user. The access request by the subject for object will be evaluated under relevant access control policy rule $a\rho \in U_{\rho}$. The considered policy rule defines the set of security function involved in authorization evaluation process. Thus the state s of the system will comprise of set of current accesses U_{CurAcc} and set of security function $U_{SF}(s)$ with respect to the considered access policy. The state can be declared as secure state if the set of current accesses are authorized under the considered policy. The set of conditions that need to be satisfied by the state can be represented as SSt_{UAM}. The secure state can be represented as $SSt_{UAM}(s)$ and set of secure state can be specified as $U_{SS|SSt} = \{s \in U_{SS} \mid SSt_{UAM}(s)\}$ .In the next section the basic model sets and various security state variables required for the development of the model is described in detail.

3.1.2 UAM Model Elementary Sets

In this section the basic element sets used for the development of the model is described in detail. The UAM model includes basic sets of entities (E) named as Subject, Objects, Environment domain, Basic Attributes, Context Attribute and Trust Attribute. The detail of various sets is defined as follow.

- **Subject:** Subject represents an entity that initiates access request to access a resource of the system.
- **Object:** Object represents an entity that is designated as a resource in the system and accessed by the other entities designated as subjects.
- **Environment Domain:** Environment domain represents the computing environment or location or surroundings in which an object resource is being accessed by the subject.
- **Basic Attribute:** Attributes are security relevant characteristics. Let $U_{Att} = \{Att_1, Att_2, ..., Att_n\}$ be a finite set of attributes. Each entity $e \in E$ is associated with finite set of attributes specified by the attribute mapping relation $R_{att}(E) \subseteq E \times U_{Att}$ where $\sum_{1}^{n} Att_i \subseteq U_{Att}$ represents the attribute profile of entity $e \in E$ .The categories of attribute include Subject attribute, Object attribute and Ubiquitous Environment domain attribute.
- **System Security Policy Base (ρ_{UAM}) :** System security policy base represents collection of finite set of security

policy rules built over security parameter associated with system entities. $\rho_{UAM} = \{ap_1, ap_2, ..., ap_n\}$ represents the policy base where ap_i is a policy rule expressed as Boolean expression which is a function of entity security descriptors. The system authorization process involves evaluation of set of applicable policies with respect to access request generated by the system for granting access to system resources. The system security policy base is collection multiple domain policy set and can be represented as follow.

$$\rho_{UAM} = \rho_{ABP} \cup \rho_{TBP} \cup \rho_{PCP} \cup \rho_{KBP} \cup \rho_{CBP} \cup \rho_{CtBP}.$$

The collection includes the following policy sets.

Attribute Based Policies (ρ_{ABP}): The Policy rules used to control the access to systems protected resources through policies expressed as Boolean function of subject, object and environmental attributes.

Trust Based Policies (ρ_{TBP}): The Policy rules used to control the access to systems protected resources through policies expressed as Boolean function of trust oriented subject, object and environmental attributes.

Performance Centric Policies (ρ_{PCP}): The Policy rules used to control the access to systems protected resources through policies expressed as Boolean function of performance centric attributes with respect to resource and environment.

Knowledge Base Policies (ρ_{KBP}): The Implicit Policy rules used to control the access to systems protected resources through policies expressed as Boolean function of facts and rules with respect to subject, resource and environment.

Case Based Policies (ρ_{CBP}): The policy rule used to control the access to systems protected resources through policies expressed as Boolean function of case parameters with respect to previous authorizes accesses of the system.

Context Based Policies (ρ_{CtBP}): The Policy rules used to control the access to systems protected resources through policies expressed as Boolean function of context oriented subject, object and environmental attributes.

- **Semantic Knowledge Base (Kb_{UAM}):** The Semantic Knowledge base is formal representation of the asserted facts and rules about the various entities of the Ubiquitous computing environment system.

  $Kb_{UAM} = \sum_{1}^{n} Kb_i$ where Kb_i is the set of assertions defined for the E_i of the Ubiquitous computing environment. To control the size of the knowledge base, only security relevant characteristics are considered. The system knowledgebase is created with the help of formal knowledge base representation tools. The Knowledgebase acts as reliable source of the information and facts about the entities of the system involved in authorization process.

After defining the basic set of the model, we define the security parameter for the UAM as follow

$$SP_{UAM} = (U_{SUB}, U_{Att}, U_{Ctx}, U_{TrAtt}, \rho_{UAM}, KB)$$

where U_{SUB} is the set of subjects, U_{Att} is the set of basic attributes, U_{Ctx} is the set of contextual attributes, U_{TrAtt} is set of trust attribute constraints, ρ_{UAM} is the finite set of system security policy rules and KB represents semantic knowledgebase comprising of defined facts and rules.

3.1.3 UAM Model State Security Variables

In this section UAM model state security variables are defined. In the context of UAM system, the state $s \in U_{SS}$ is a tuple.

$$s = (U_{CurAcc}, U_{Sub}, SF_{att}, SF_{tr}, SF_{Ctx}, U_{AuthStateP}, U_{APl}, U_{AccPerm}, SF_{KB}).$$

The state of the system describes the state variables that capture the security relevant information of the authorization process. Such Security relevant state variables can be termed as Security Function (SF_{UAM}) .The security function associates the security information to the entities of the system. With the change of state of the system, the security function may vary. In the following we describe all the components used in state definition.

- **Current Access Set:** With respect to the current state of the system, U_{CurAcc} represents the set of current authorized accesses in the system.
- **Subject Attribute Relation**: This relation represents the list of subject attribute in a particular state associated with an entity subject $Sub_i \in U_{SUB}$. The Subject Attribute relation is many to many mapping and can be represented as $SAR_{att} \subseteq U_{Sub} \times U_{SubAtt}$.
- **Object Attribute Relation**: This relation represent the list of object attribute in a particular state associated with an entity object $Obj_i \in U_{OBJ}$. The Object Attribute relation is many to many mapping and can be represented as $OAR_{att} \subseteq U_{Obj} \times U_{ObjAtt}$
- **Env-Dom Attribute Relation**: This relation represent the list of environment attributes in a particular state associated with an entity environment domain $Ed_i \in U_{EnvDom}$. The Environment Attribute relation is many to many mapping and can be represented as $EAR_{att} \subseteq U_{EnvDom} \times U_{EnvAtt}$
- **Entity Context-Attribute Relation:** The Contextual Attribute mapping relation can be represented as $ECR_{Ctx} \subseteq E \times U_{CtxAtt}$ where E is set of all entities comprises of subject, objects and environment domain.
- **Entity Trust-Attribute Relation:** The Trust Attribute mapping relation can be represented as $ETR_{Tr} \subseteq E \times U_{TrAtt}$ where E is set of all entities comprises of subject, objects and environment domain.
- **Attribute based Security Function (SF_{att}):** This class of function is a set comprising of attribute based security function.
- **Context Based Security Function(SF_{Ctx}):**Security Function (SF_{Ctx}) is a set of security relation comprising of context based security function.
- **Trust Based Security Function (SF_{Tr}):** Security Function (SF_{Tr}) is a set of security relation comprising of trust based security function.
- **Authorization Policy (APl) :** The Authorization Policy represents the conditions under which access triple

$\langle Sub, Obj, AccOp \rangle$ is allowed. The access conditions are expressed as Boolean function of entity security descriptors that include attributes, context, trust, knowledgebase facts/assertions and policy rules. The access is granted if function evaluates to true, otherwise the access is denied. Formally authorization policy can be represented as follow

$$APl : Access\_Allowed(< Sub, Obj, AccOp >) \leftarrow f(PCond_1, PCond_2, ..., PCond_n)$$

In the above function each policy condition $PCond_i \in AuthCd(apl)$ is a Boolean attribute or set of multiple attributes represented as a Boolean expression called $< PolicyRule >$. The set Authorization condition $AuthCd(apl) = \{PCond_1, PCond_2, ..., PCond_n\}$ is a finite set of Boolean functions where $PCond_i \rightarrow \{T, F\}$.If any of the policy condition returns False, the access is blocked. If all the policy conditions returns true then access is allowed and subject can perform specified action on a resource object as per authorization assignment $AuthStateP$

- **Authorization Policy Base** (U_{APl}) **:** Authorization Policy base consists of set of authorization policy rules that provide the protection of all system resources within the ubiquitous computing environment. The Authorization process of the system involves evaluation of applicable policy rules from the policy base. Authorization Policy base can be represented as follow

  $$U_{APl} = \{apl_1, apl_2, ..., apl_n\}$$

- **Authorization Evaluation Function:** The Authorization evaluation function is represented as $f_{AE}(ar) : P \rightarrow D$ where the domain $P = \{apl_1, apl_2, ..., apl_n\}$ is as set of applicable Access Control Policies from policy base U_{APl} with respect to the Access request ar. The applicable access control policies from Policy base are applied to the Access request ar and decision with respect to each access control policy is combined by Authorization evaluation function using combination algorithm $CA = \{apl_1 \oplus apl_2 \oplus ..., apl_k\}$ to conclude the final decision that whether the access request is permitted or denied under the domain $D = \{permit, deny\}$.
- **Knowledge Base** $f_{Kb}(E)$ **:** The Knowledge Base function is used to infer the knowledge about the system entities involved in the authorization process. The function $f_{Kb}(E) : A \rightarrow C$ has an antecedent domain A and a consequent domain C. The Knowledge base $Kb_i \in KB$ represents set of assertions (facts and rules) for entity E_i The antecedent and consequent domain consists of conjunction of entity property represented as predicate. If the entire component in the antecedent domain of relation is true then consequent of the relation must also be true.
- **Access Permission:** The access permission allows subject with specific attributes to perform specific operation on an object. The access permission can be represented as $p(\wp(Obj, AccOp), < rsp_1 >, < rsp_2 >, ..., < rsp_n >)$, where $< rsp >$ is a reference to the security policy parameters defined in the model. The referenced security policy parameters can be access control policy, knowledge based rules, facts, attribute constraints or any other constraint defined in the model. The standard parameters can be defined as follow.

1. $\wp(Obj, AccOp)$ Represents allowed set of Access Operation on Resource Object.
2. $rsp < apl >$ Represents Access control policy along with required set of Constraints/Conditions of Policy.
3. $rsp < SAR_{Att} >$ Represents required set of attributes of Subject for permission p
4. $rsp < OAR_{Att} >$ Represents required resource object attribute profile.
5. $rsp < p >$ Represents the security parameters that define the applicability of access permission p.

The access permission set can be defined in terms of above components as $U_{AccPerm} = \wp(U_{Sub} \times U_{AccOp} \times U_{Obj} \times rsp)$ .The permission specifies the prerequisites that subject should satisfy before being allowed to exercise the set of privileged operation over resource object. The permission set are derived from Access Control Policies defined as policy rules for protection of system resources.

- **Authorized State Permission** $(U_{AuthStateP})$ **:** The Authorization State Permission set is used represent authorized permissions that a subject of the system is authorized to perform on an object in a particular state s. The authorization is decided based on the subject, object, environment descriptors, security constraints and rules of the authorization system model. The authorization is expressed in terms of entity descriptors only. Formally authorized state permission set can be represented as follow.

  $$U_{AuthStateP}(s) = \{p_1, p_2, ..., p_n\} where\ p \subseteq U_{AccPerm}$$

  This relation represents the list of permission in a particular state. As per association user with a particular active set of attributes profile from SAR_{Att} under specific model constraints and policy rules $APl < PolicyID, PCond >$ will be able to exercise the access privilege under permission p on a resource object represented as $Obj < OAR_{Att} >$ provided user's current profile set satisfies the active set criteria defined under access permission p associated with resource object.

4. UAM MODEL SECURE STATE

After defining basic model sets and relation for the UAM model, in this section the secure state for the UAM model is described. For a given state $s \in U_{ss}$, the secure state invariant can be represented as $SSt_{UAM}(s)$ .The set of secure states can be defined as $U_{SS|SSt_{UAM}} = \{s \in U_{SS} \mid SSt_{UAM}(s)\}$ .To formulate the criteria for secure state $SSt_{UAM}(s)$ we need to identify and consider all the security properties that must hold for state confirmation as secure state. In the following we define security properties that form the criteria for secure state under UAM model. The $SSt_{UAM}(s)$ holds iff

1. $s \in U_{ss}$ satisfies Attribute based constraints under Current Knowledgebase i.e.

- Subject Attribute Assignment Constraint

  - $\forall Sub \in U_{SUB}$,
    $assigned\_Att\_Sub(Sub) \subseteq U_{SAtt}(Sub, SAR_{Att})$ where ,

given a subject $Sub \in U_{Sub}$, $U_{SAtt}(Sub, SAR_{Att})$ is the set of attributes that Sub can activate according to SAR_{Att} .

- Object Attribute Assignment Constraint
  - $\forall Obj \in U_{OBJ}$, $assigned\_Att\_Obj(Obj) \subseteq U_{OAtt}(Obj, OAR_{Att})$ where , given a object $Obj \in U_{Obj}$, $U_{OAtt}(Obj, OAR_{Att})$ is the set of attributes that Obj can activate according to OAR_{Att} .
- Environment Domain Attribute Assignment Constraint
  - $\forall Ed \in U_{EnvDom}$, $assigned\_Att\_Ed(Ed) \subseteq U_{EAtt}(Ed, EAR_{Att})$ where , given a subject $Ed \in U_{EnvDom}$, $U_{EAtt}(Ed, EAR_{Att})$ is the set of attributes that Ed can activate according to EAR_{Att} .

2. $s \in U_{ss}$ satisfies Contextual Attribute constraints under Current Knowledgebase.

- Entity Contextual Attribute Assignment Constraint
  - $\forall e \in E$, $assigned\_CtxAtt\_Entity(e) \subseteq U_{ECtxAtt}(e, ECR_{Ctx})$ where , given a subject $e \in E$, $U_{ECtxAtt}(e, ECR_{Ctx})$ is the set of attributes that e can activate according to ECR_{Ctx} .

3. $s \in U_{ss}$ satisfies Trust Attribute constraints under Current Knowledgebase

- Entity Trust Attribute Assignment Constraint
  - $\forall e \in E$, $assigned\_TrAtt\_Entity(e) \subseteq U_{ETrAtt}(e, ETR_{Tr})$ where , given a subject $e \in E$, $U_{ETrAtt}(e, ETR_{Tr})$ is the set of attributes that e can activate according to ETR_{Tr} .
  - $\forall e \in E$, $TrAtt(e) \subseteq U_{ETrAttV}(e, AVR_{TrAtt})$ where, given a subject $e \in E$, $U_{ETrAttV}(e, AVR_{TrAtt})$ is the set representing range of values that e can activate for $TrAtt$ according to AVR_{TrAtt} .

4. Resource Access Constraint :

$$\forall Sub \in U_{Sub}, \forall Obj \in U_{Obj}, \forall AccOp \in U_{AccOp}$$
$$(Sub, Obj, AccOp) \in U_{CurAcc} \Rightarrow (Obj, AccOp) \in U_{SPerm}(Sub, U_{AuthStateP})$$ where

$U_{SPerm}(Sub, U_{AuthStateP})$ is set of permissions associated to a subject $Sub \in U_{Sub}$ as per $U_{AuthStateP}$.

After defining system constraints, for a given state s, the specification of the secure state $SSt_{UAM}(s)$ among set of states can be stated as follow.

Definition 1: A state $s \in U_{SS}$ is secure and holds secure state invariant if and only if.

1. s satisfies Attribute Assignment Constraint.
2. s satisfies Contextual Attribute Assignment Constraint.
3. s satisfies Trust Attribute Assignment Constraint.
4. s satisfies Resource Access Constraints.

The above definition is represented formally as an invariant $SSt_{UAM}(s)$ .The invariant $SSt_{UAM}(s)$ expresses relationship between values of the state security variables that must always be maintained during state transition from one state to another state.

5. UAM ACCESS CONTROL POLICY

Based on the security parameters for Ubiquitous computing environment, The UAM access control policy for Ubiquitous computing environment can be defined as follow.

$$AuthPolicy_{UAM}[SP_{UAM}] = \{ < U_{SUB}, U_{OBJ}, U_{AccOp} >, U_{SS}, SSt_{UAM} \}$$

where $< U_{SUB}, U_{OBJ}, U_{AccOp} >$ represents non empty set of Subject, Object and Access Operation respectively, U_{SS} is set of states of the system covered under policy $AuthPolicy_{UAM}$, SSt_{UAM} is the functional invariant that must be evaluated for each system state to qualify as secure state.

6. FORMAL UAM MODEL

After defining the conditions for the secure state and based on the Access control Policy $AuthPolicy_{UAM}[SP_{UAM}] = \{ < U_{SUB}, U_{OBJ}, U_{AccOp} >, U_{SS}, SSt_{UAM} \}$ with security parameter SP_{UAM} , the Ubiquitous Access Control Model $M_{UAM}(SP_{UAM})$ is defined as a tuple

$$M_{UAM}(SP_{UAM}) = \{AuthPolicy(SP_{UAM}), U_{AR}\}.$$

A ubiquitous access control model $M_{UAM}(SP_{UAM})$ is used to specify secure state of the system based on the access control policy. To implement the security model initial system state, state transition function and set of system management functions need to be defined.

6.1 UAM Management Operations

The UAM Management Operation set MOp_{UAM} is a collection of Administrative and User operations for management of the UAM model components. These functions control the changes of model state variables as per constraints defined under system operations. The set of operation can be represented as $MOp_{UAM} = MOp_{Ad\min} \cup MOp_{Usr}$ where $MOp_{Ad\min}$ represents set of Administrative operation and MOp_{Usr} represents set of User Operations. Administrative operations are used for the management of UAM State security variables and User Operations are used for Access Management as per the need of users for UAM model. We consider set of user access request operation $U_{AR} = \{ar_1, ar_2, ..., ar_n\}$ as subset of System user operations. The following figure summarizes the set of Administrative Operations and set of User Operations.

6.1.1 System Administrative Functions

In this section the system administrative functions for performing administrative operations are described.

1. Add Subject: This is an operation used for creation of a new subject. The operation is allowed with precondition that the new subject is not the member of the U_{SUB} data set. After successful completion of operation new subject is created and the U_{SUB} data set with other relevant functions are updated. The formal definition of the function along with security conditions is described as follow.

$AddSubject(Sub : U_{Sub})\{$
$If\ \ Sub \notin U_{Sub}$
$Then\ U'_{Sub} = U_{Sub} \cup \{Sub\}$
$Set\ assigned\_Att\_Sub' = assigned\_Atts\_Sub \cup \{Sub \mapsto \phi\}$
$Set\ assigned\_CtxAtt\_Entity' = assigned\_CtxAtt\_Entity \cup \{Sub \mapsto \phi\}$
$Set\ assigned\_TrAtt\_Entity' = assigned\_TrAtt\_Entity \cup \{Sub \mapsto \phi\}\}$

2. Add Subject-Attribute: This is an operation used for creation of a new attribute of a subject. The operation is allowed with precondition that the new attribute is not the member of the U_{SubAtt} data set. After successful completion of operation new subject attribute is created and the U_{SubAtt} data set with other relevant functions are updated. The formal definition of the function along with security constraints is described as follow.

$AddSubjectAttribute(SubAtt : U_{SubAtt})\{$
$If\ \ SubAtt \notin U_{SubAtt}$
$Then\ U'_{SubAtt} = U_{SubAtt} \cup \{SubAtt\}$
$Set\ assigned\_Sub\_Att' = assigned\_Sub\_Att \cup \{SubAtt \mapsto \phi\}$
$Set\ SubAtt' = SubAtt \cup \{v \mapsto \phi\}$
$\}$

3. Authorized State Permission Assignment : This is an operation used to define available authorized permission p in a particular state.

$AssignAuthStateP(Obj : U_{Obj}; AccOp : U_{AccOp}; p \in U_{AccPerm})$
$\{$
$p \in U_{AccPerm}; Obj \in U_{Obj}; AccOp \in U_{AccOp};$
$For\ each\ \ p(Obj, AccOp) \in U_{AccPerm}$
do
$for\ each\ \ p < rp > \in p$
$if\ \ p < rp > \rightarrow \{True\}, then$
$U'_{AuthStateP} = U_{AuthStateP} \cup \{p(Obj, AccOp)\}$
$Else$
$if\ \neg p < rp >$
$U'_{AuthStateP} = U_{AuthStateP} / \{p(Obj, AccOp)\}$
$\}$

The operation is allowed with the precondition that the permission p is a member of the Access Permission data set $U_{AccPerm}$, the subject is a member of U_{Sub} data set. After successful completion of the operation the $AuthStateP$ data set with other relevant functions are updated. The formal definition of the function along with security constraints is described as above.

6.1.2 System Users Functions

1. Authorization Request Evaluation Function: This is an operation used to evaluate user request $ar(Sub, Obj, AccOp) \in U_{AR}$ against available authorized permissions $p \in U_{AuthStatep}$ in a particular state. This function returns a Boolean value in a parameter Eval.If the value is true, then the subject is allowed to perform a specified Access Operation on a resource object. The formal definition of the function along with security constraints is described as follow.

$AuthEval(Sub : U_{Sub}; Obj : U_{Obj}; AccOp : U_{AccOp}; Eval)\{$
$Sub \in U_{Sub}; Obj \in U_{Obj}; AccOp \in U_{AccOp};$
$If\ (Sub, Obj, AccOp) \in U_{AuthStateP}$
$Eval = \{True\}$
$Else$
$Eval = \{False\}$
$\}$

6.2 UAM State Transition Function

The state transition function represents set of functions when applied to system, results in transition from one state to another state. In order to implement secure model, we need to ensure that all state transition maintains secure state. Let us define a set U_{IS} as set of initial states .Let $MOp_{Usr} \in MOp_{UAM}$ be the user access operation and a set U_{SS} as set of system states. The state transition function $tf(MOp, s, s') : Mop \times U_{SS} \rightarrow O(Mop) \times U_{SS}$ allows the transition from one state to another state where $O(Mop)$ is the outcome of the system user operation $Mop \in MoP_{UAM}$ obtained by applying the member function over a state $s \in U_{SS}$. To maintain secure state, the state transition function when applied to the state s should result in state such that the secure state invariant holds over derived state. This security condition is represented through the following theorem

$SSt(s) \wedge tf \Rightarrow SSt(s')$ where s' represents the derived state.

The above condition will hold when all the member operations of the state transition function satisfy the security conditions defined for each of the member operations.

6.3 UAM Initial State

The initial state of the system can be defined as state S_0. In order to represent the S_0 state we consider the state s with set of initial values. This can be represented as follow

$s_0 = (U_{CurAcc}, U_{Sub}, SF_{att}, SF_{tr}, SF_{Ctx}, U_{AuthStateP}(s), U_{APl}, U_{AccPerm}, SF_{KB})$ where each state variable can be initialized with null value. For secure system

we should ensure that initial state is also a secure state i.e. initial state should also satisfy the secure state invariant $SSt_{UAM}(s_0)$. The case in which the initial state can be assumed to be secure state is with all its state variables initialized to null value i.e. $U_{CurAcc}=\phi, U_{Sub}=\phi, SF_{att}=\phi, SF_{tr}=\phi, SF_{Ctx}=\phi,$ $U_{AuthStateP}(s)=\phi, U_{APl}=\phi, U_{AccPerm}=\phi, SF_{KB}=\phi$

In this case the condition $U_{Sub}=\phi,$ is considered sufficient for the state to be declared as secure state irrespective of the status of the other state variables .In order to make transition from initial state to next secure state we need administrative function to add subject into the system.

In the case of initial state where $U_{Sub}\neq\phi,$ the state s_0 is said to be secure if it comply with all the conditions of as defined for SSt_{UAM} invariant.

7. UAM IMPLEMENTATION

The Authorization model defines the security criteria for the implementation of the secure systems. The security criteria are represented through Authorization policy that includes the secure state invariant. In order to implement the model we defined the required components as part of model development process in previous sections. These components are Model State Variables, Model Initial State (s_0), Model State Transition Function (tf_{UAM}), Model System Management Operation (MOp_{UAM}) and Model State Security Invariant (SSt_{UAM}). The model Implementation can be represented as (tf_{UAM}, U_{IS}) where tf_{UAM} represents the transition function

The transition function allows the system to transit from one state to the another state starting from initial secure state. Let the set of reachable state can be represented as $U_{RS|tf}\in U_{SS}$ after the application of transition function tf from current set of states U_{CS}. The generic transition function $tf(ar,s,s')$ for the model for the Access request $ar(Sub,Obj,AccOp)$ can be defined as follow in Figure 2.

$$tf(ar,s,s')=\begin{cases} s'=s\cup(Sub,Obj,AccOp) \\ if\,(ar=Allowed)\wedge(s'\in SSt_{UAM}) \\ \\ s'=s\cup\{\phi\} \\ if\,(ar=Denied)\wedge(s'\in SSt_{UAM}) \\ \\ s(otherwise) \end{cases}$$

Fig 2: The Generic Transition function.

The generic definition describes the cases of Access request for $\{Allowed, Denied\}$ values by using the state security invariant. For 'Allowed' case, the access request is added to the state s, subject to the constraints of state security invariant. For 'Denied' case, the access request is removed from the state s, subject to the constraints of state security invariant. Based on generic definition we can define now transition function UAM for the following state s.

$s=(U_{CurAcc}, U_{Sub}, SF_{att}, SF_{tr}, SF_{Ctx},$ $U_{AuthStateP}(s), U_{APl}, U_{AccPerm}, SF_{KB})$

$$tf_{UAM}(ar,s,s')=\begin{cases} s'=s(U_{CurAcc}\cup\{(Sub,Obj,AccOp)\},\ldots) \\ if\,(ar=Allowed)\wedge(s'\in SSt_{UAM})\wedge \\ (Obj,AccOp)\in U_{SPerm}(Sub,U_{AuthStateP}) \\ \\ s'=s(U_{CurAcc}\setminus\{(Sub,Obj,AccOp)\},\ldots) \\ if\,(ar=Denied)\wedge(s'\in SSt_{UAM}) \\ \\ s(U_{CurAcc},\ldots)otherwise \end{cases}$$

Fig 3: The UAM Transition function.

Definition 2: The system implementing the model $M_{UAM}(SP_{UAM})$ is said to be secure if and only if.

1. The initial state S_0 is secure state.
2. The transition function $tf_{UAM}(ar,s,s')$ satisfies the model constraints defined for the member access request operation $ar\in U_{AR}$ where $U_{AR}\subseteq MOp_{UAM}$.
3. All the states reachable from the initial state are also secure. This can be formally represented as follow.

$(tf_{UAM}(ar,s,s'),U_{IS})\cap AuthPolicy_{UAM}(SP_{UAM})\Rightarrow$ $U_{RS|tf}\subseteq U_{SS|SSt_{UAM}}$

where $s\in U_{IS}$ be secure initial state and on application of transition function $tf(ar,s,s')$ under the access control policy $AuthPolicy_{UAM}(SP_{UAM})$ returns state that always belong to set of secure state.

The implementation (tf_{UAM}, U_{IS}) for a given model $M_{UAM}(SP_{UAM})=\{AuthPolicy(SP_{UAM}), U_{AR}\}$ is said to be model compliant iff it satisfies all the model constraints defined under above definition. The above definition describes formally the security requirements for the implementation of the Authorization model for secure ubiquitous computing environment.

8. CONCLUSION

In this paper the Ubiquitous Authorization model that provides a secure authorization framework for implementation of secure authorization service in a ubiquitous computing environment is developed. A formal approach based on state machine approach is adopted to design a flexible and scalable model to support intelligent authorization process in ubiquitous computing environment. In the proposed model, security parameters have been defined to address the security needs of the ubiquitous computing environment. Based on the security parameters, the Authorization policy for the system is formulated. Also system security invariant is defined to ensure the protection of system states. After defining the basic components i.e. security parameters, security invariant and Authorization policy, a formal Ubiquitous Authorization Model is developed. The proposed model ensures the protection of system resources of the Ubiquitous computing system and provides secure mechanism for implementing authorization service.

9. REFERENCES

[1] M. Weiser 1991, "The Computer for the Twenty-First Century," in *Scientific American*, vol. 265, 1991, pp. 94-104.

[2] M. Weiser 1993, Ubiquitous Computing, Computer, v.26 n.10, p.71-72.

[3] Lampson, Butler W. 1971. "Protection". Proceedings of the 5th Princeton Conference on Information Sciences and Systems. p. 437.

[4] David F. Ferraiolo and D. Richard Kuhn 1992, "Role-Based Access Controls," *Proceedings of the 15th NIST-NSA National Computer Security Conference,* Baltimore, Maryland, October 13-16.

[5] Ravi S. Sandhu 1993. Lattice-based access control models. IEEE Computer.

[6] Ravi S. Sandhu and P. Samarati 1994. Access control: Principles and practice. IEEE Com. Mag.

[7] Antonio Corradi, Rebecca Montanari, and Daniela Tibaldi. 2004. Context-Based Access Control Management in Ubiquitous Environments. In *Proceedings of the Network Computing and Applications, Third IEEE International Symposium* (NCA '04). IEEE Computer Society, Washington, DC, USA, 253-260

[8] Lin L. and Tianjie C., 2008. A Flexible, Autonomous and Non-redundancy Access Control for Ubiquitous Computing Environment. In *Proceedings of the 2008 International Symposium on Information Science and Engieering - Volume 01* (ISISE '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 446-450.

[9] Hung L. X., Riaz A. S., Hassan J., Raazi S. M., Yuan W., Ngo T. C., Truc P. T., Lee S., Lee H., Yuseung S., and Miguel F. 2009. Activity-oriented access control for ubiquitous environments. In *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference* (CCNC'09). IEEE Press, Piscataway, NJ, USA, 1402-1406.

[10] Sejong O., 2010. New role-based access control in ubiquitous e-business environment. *J. Intell. Manuf.* 21, 5 , 607-612.

[11] Manachai T. and Indrakshi R., 2011. On the formalization and analysis of a spatio-temporal role-based access control model. *J. Comput. Secur.* 19, 3 , 399-452.

[12] Sigrid S. and Strembeck M. 2012. Modeling Context-Aware RBAC Models for Business Processes in Ubiquitous Computing Environments. In *Proceedings of the 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing* . IEEE Computer Society, Washington, DC, USA, 126-131.