AC4E: An Access Control Model for Emergencies of Mission-Critical Cyber-Physical Systems

Dong Chen[1], Guiran Chang[2] and Jie Jia[1]
[1] School of Information Science and Engineering, Northeastern University
Shenyang, China
[2] Computing Center, Northeastern University
Shenyang, China
[e-mail: chend.2008@gmail.com, jiajie@ise.neu.edu.cn, chang@neu.edu.cn]
*Corresponding author: Dong Chen

Received May 11, 2012; revised July 4, 2012; accepted August 16, 2012; published September 26, 2012

Abstract

Access control is an essential security component in protecting sensitive data and services from unauthorized access to the resources in mission-critical Cyber-Physical Systems (CPSs). CPSs are different from conventional information processing systems in such that they involve interactions between the cyber world and the physical world. Therefore, existing access control models cannot be used directly and even become disabled in an emergency situation. This paper proposes an adaptive Access Control model for Emergences (AC4E) for mission-critical CPSs. The principal aim of AC4E is to control the criticalities in these systems by executing corresponding responsive actions. AC4E not only provides the ability to control access to data and services in normal situations, but also grants the correct set of access privileges, at the correct time, to the correct set of subjects in emergency situations. It can facilitate adaptively responsive actions altering the privileges to specific subjects in a proactive manner without the need for any explicit access requests. A semiformal validation of the AC4E model is presented, with respect to responsiveness, correctness, safety, non-repudiation and concurrency, respectively. Then a case study is given to demonstrate how the AC4E model detects, responds, and controls the emergency events for a typical CPS adaptively in a proactive manner. Eventually, a wide set of simulations and performance comparisons of the proposed AC4E model are presented.

Keywords: Cyber-physical systems, access control, security, emergencies

This work is supported by the National Natural Science Foundation of China under Grant No. 60903159 and No. 61173153.
http://dx.doi.org/10.3837/tiis.2012.09.003

1. Introduction

Cyber-physical systems will have to support various communication technologies and integrate different devices. A typical Cyber-Physical System (CPS) consists of two major components, physical processes and an intelligent cyber system [1][2]. Physical processes are usually monitored and controlled by the cyber system which is often a networked system of several tiny smart devices with sensing, computing and wireless communication capabilities [3][4]. The emergence of CPS applications have effect on the revolution, including assisted living, intelligent traffic control and safety, energy conservation, enviromental control, instrumentation and avionics. Among the applications, some CPS systems are mission-critical applications, such as Medical CPSs (MCPSs) [5][6], Smart Grid CPSs (SGCPSs) [7] and etc.. While incorporating cyber systems and networks, MCPSs and SGCPSs will be exposed to a wide range of security threats [8][9]. In these mission-critical domains, emergencies [10][11] have become one of the biggest threats to the access security of the CPSs.

Access control [12] is a process of limiting access to the resources of a system only to authorized programs, processes, or other systems. Different access control models have been proposed over the years. Among the existing models, Role Based Access Control (RBAC) [12] is the most influential one. Furthermore, many RBAC based models, such as TrustBAC [13], OS [14], CBAC [15], CAAC [16] and etc., have been proposed. Access control is a serious problem because CPSs involve interactions between a great number of entities which may span different organizational boundaries [5]. However, the existing models provide access services statically and explicitly in a reactive manner, and lack consideration of physical contexts during the designing period of control policies. In case of emergency situations, traditional models cannot provide appropriate alternate access privileges to execute responsive actions to control the critical events dynamically and to keep the system stable in a proactive manner. Therefore, novel access control models have to be proposed for CPSs before they are deployed widely.

The contribution of this paper can be categorized as follows: (1) Analysis of access control challenges of mission-critical CPSs; (2) Concepts of emergency degree and emergency dependency, and the classification of emergency dependency; (3) Evaluation metrics of emergency degree for CPSs; (4) A novel adaptive and proactive Access Control model for Emergencies (AC4E) of mission-critical CPSs; (5) A semiformal validation of the AC4E model with respect to responsiveness, correctness, safety, non-repudiation and concurrency, respectively; (6) A case study to demonstrate how the AC4E model detects, responds and control multiple emergencies for a typical CPS adaptively in a proactive manner.

The remainder of the paper is organized as follows. We give an overview of related influential works in Section 2. In Section 3, the definitions of emergency dependency and emergency degree are given, and the relationship between different emergencies is discussed. In Section 4, an access control model, AC4E, for emergency situations in mission-critical CPSs is presented in detail, including the design goals, primitives, emergency detection, action generation, responsive action generation and execution schemes. In Section 5, a semiformal validation of the AC4E model is presented. A case study is presented also to demonstrate how AC4E model detects, responds and controls the emergency events adaptively in a proactive manner in Section 6. In Section 7, we give several groups of performance comparisons of AC4E model. Eventually, we conclude this paper and discuss future works in Section 8.

2. Related Work

Access control is an essestial tool in preventing unauthorized access to the available sensed data by the underlying wireless networks in CPSs. Although different access control models have been proposed over the years, role-based access control (RBAC) [12] is gradually emerging as the standard and the most influential one for access control. S. Chakraborty etc. [13] propose a trust based access control model which extends the conventional RBAC models with the notion of trust levels. I. Ray and M. Toahchoodee [17][18] propose a formal spatio-temporal model based on RBAC model. The association of each component of RBAC with spatio-temporal information and formalization by enumerating the constraints are shown. S. C. Yu etc. [19] propose a fine-grained distributed data access control schem, specially tailored for distributed wireless networks of CPSs based on continuous observation of the inherent nature of the sensor data. S. Misra and A. Vaish [20] propose a novel reputation-based role assignment for RBAC to evict highly non-cooperative and malicious nodes from wireless networks. However, these models have often been found to be inadequate for scalable and mission-critical CPSs where the user population is dynamic and the identities of all users are not known at all before deployed to physical environments.

Emergency is defined as the effect of a series of events in physical world [21], which can cause the system to enter unstable states. D. Povey [14] proposes an optimistic access control scheme where enforcement of rules is retrospective. Under emergency situations or exceptional circumstances, the legitimate access requirements should be relaxed. C. K. Georgiadis etc. [22] use the integration of contextual information with team-based access control to provide access control for collaborative activities best accomplished by teams of users. A. Corradi etc. [15] propose an access control model which proposes the adaption of context as a first-class design principle to rule access to resources. S. K. S. Gupat etc. [21] introduce the concept of criticality, which measures the level of responsiveness in taking such actions and present an access control model to aid the pervasive systems to handle critical events. G. W. Wu [23] etc. propose a new access control scheme which provides an adaptive access control policy specifically to address the multiple emergency management problem in CPSs. S. Yu, K. Ren and W. Lou [24] propose a distributed data access control scheme which is able to fulfill fine-grained access control over sensor data and is resilient against strong attacks.

However, these schemes are statically defined before the CPSs deployed, and cannot be adjusted according to the change of the system environment dynamically. The important task is to identify what types of access control policies are suitable for CPSs. Traditional schemes cannot provide proper privileges to execute the responsive actions to avoid the failure of the system, especially under emergency situations. Most of them can only handle one emergency at one time and lack of consideration of environmental contexts and feedback from the environment around the CPSs when making access decisions. In mission-critical CPSs, such as Medical Device Plug-and-Play (MDPnP) [25][26] systems and NUAV surveillance systems [27], one of the key characteristics of CPSs is their close interaction with their environment [28]. Obviously, the physical environment is a key factor of the mission-critical CPSs when making access control decisions. Therefore, we propose an adaptive and proactive access control model to meet the security requirements of mission-critical CPSs.

3. Emergencies in Mission-critical CPSs

3.1 Definition of Emergency

CPSs represent a new breed of emerging systems where cyber world, physical world and interactions between them are involved. The resulting cyber-physical coupling is tightly controlled, tunable, precise and predictable [29][30]. The CPS revolution promises scientific and engineering breakthroughs to address such challenges and problems. Emergencies, also called criticalities, have been defined as the sequences of specific critical events in a smart system [16]. Emergency management in mission-critical CPSs is more challenging than other traditional systems. Besides critical events of cyber processes, critical events of physical environment and integrations between cyber processes and physical environment can also make CPSs unstable. Especially, in mission-critical domains, such as electric grid, traffic control systems, medical devices, air pollution control, and unmanned vehicles, etc., emergencies have become one of the biggest threats to the security of the CPSs. Therefore, a new definition has to be proposed.

Definition 1. An Emergency in mission-critical CPSs is an unstable system state caused by critical events of the cyber processes, accidents in physical environment or errors and conflicts of the interactions between the cyber system and its surrounding environment.

3.2 Emergency Dependency

In mission-critical CPSs, emergencies usually occur in groups. Furthermore, different emergencies may have tight relationships between them. Some emergencies may need the same subjects to perform responsive actions, and even may have conflicts on the location or temporary privilege duration. In order to illustrate and deal with these emergencies, several definitions are given as follows.

Definition 2. Emergency Dependency denotes the compactness between one emergency and any other emergencies at the same specific time interval.

Emergency dependency can be divided into physical dependency, cyber dependency, and cyber-physical interaction dependency.

Definition 3. Physical Dependency indicates some of the responsive actions required by different emergencies may need the same subject or environmental context to execute. It also means that different emergencies may have physical conflicts or inclusion relationships in the same emergency set.

Definition 4. Cyber Dependency indicates that some of the responsive actions required by different emergencies are generated based on the same cyber properties.

Definition 5. Cyber-Physical Interaction Dependency indicates that some of the responsive actions required by different emergencies are generated based on the same bidirectionally coupled interactions between the cyber system and the physical environment.

3.3 Emergency Evaluation

In the real environments of mission-critical CPSs, several emergencies may occur in a specific system in a time interval. Therefore, a control policy has to be presented to handle these emergency events and keep the system stable. Things to start with are to evaluate the emergencies, and provide priority sequences and parallel sequences for the access control.

Definition 6. *Emergency Degree (ED)* is an evaluation of the responsive level required for taking appropriate actions to handle the corresponding critical events and to make the CPSs change to the normal state.

In this paper, *ED* can be evaluated by:

$$E_{ed} = \left[o_{\min}, \frac{1}{n} \cdot \sum_{i=1}^{n} e_i, \frac{1}{n} \cdot \sum_{i=1}^{n} p_i \right] \cdot \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}, \alpha + \beta + \gamma = 1. \quad (1)$$

where $o_{\min}$ is the minimum time window of opportunities, and e_i denotes the *i*-th execution time to control the emergency in the past transactions. p_i denotes the probability of handling the emergency successfully. We use α, β, γ to describe the corresponding weight of the three values discussed above to evaluate the *ED* of an emergency.

4. Access Control for Emergencies of Mission-critical CPSs

4.1 System Model

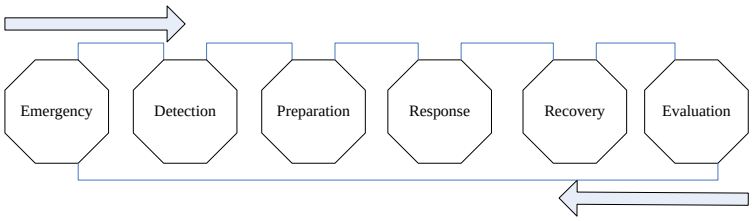
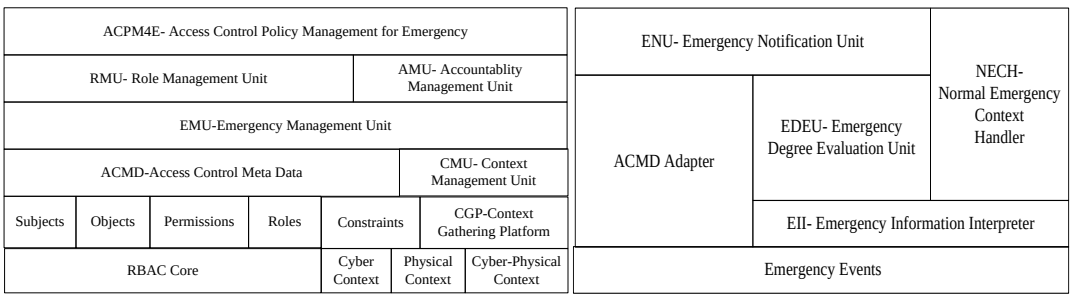


Fig. 1. Emergency management process of AC4E for CPSs

As shown in **Fig. 1**, the emergency management process of AC4E for mission-critical CPSs usually has five phases, including emergency detection, preparation, response, recovery and evaluation. The detection phase is responsible for detecting emergencies which may occur in the CPSs periodically. In the preparation phase, the optimal responsive action path is selected after the detection of emergencies. Then in the response phase, temporary roles and privileges are assigned to the corresponding subjects, and the specific responsive actions are executed to bring the emergencies under control, so to protect the system. In the recovery phase, recovery efforts are executed to make the system working correctly and stable. The last phase is responsible for the evaluation of the emergency and the corresponding responsive actions which are involved in this management cycle. The evaluation information can be used in the next optimal responsive action path selections.



(a) System structure of the AC4E model (b) System structure of EMU

Fig. 2. AC4E model for emergency management in mission-critical CPSs

In order to realize emergency management, we propose an AC4E model which is designed to handle emergencies and keep the system stable and healthy. The system structure of the AC4E model is shown in **Fig. 2(a)**. The lowest layer is the data layer, which is responsible for collecting information from RBAC-Core, cyber processes, physical environment and interactions between cyber systems and the physical environment. ACMD (Access Control Meta-Data) abstracts RBAC-Core data and the constraints between different roles, while the context information is gathered by the CGP (Context Gathering Platform) and managed by the CMU (Context Management Unit). Base on the ACMD and context data from CMU, the RMU (Role Management Unit) then enforces the appropriate access privileges using EMU (Emergency Management Unit), as shown in **Fig. 2(b)**. AMU is responsible for recording all events or transactions in the CPSs for accountability, while the ACPM4E (Access Control Policy Management for Emergencies) uses the underlying infrastructure to implement the access control and administrative policies. The EII (Emergecy Information Interpreter) can monitor the status information and context information of the CPSs in a proactive manner, and then detect the occurrence of emergencies. The EDEU (ED Evaluation Unit) is responsible for the evaluation of the ED depending on the specific context of the emergency detected. The ENU (Emergency Notification Unit) component notifies other components of the system the emergency type and its corresponding ED, and then the system are moved to AC4E working mode. The corresponding policies are provided by ACPM4E, which are the interface for ACMD. The NECH (Normal Emergency Context Handler) is employed to handle the normal contexts not involved in the emergency situations.

4.2 Design Goals

The principal characteristics that AC4E model should have are adaptiveness and proactivity. Therefore, we present five evaluation metrics to characterize the two design goals. They are: (1) Responsiveness: When an emergency occurs, the system can detect it immediately and notify the selected subjects their changed access privileges; (2) Correctness: The subjects get emergency roles and the alternate privileges if and only if the system is running under emergency situations; (3) Safety: The temporary roles and privileges can only be used under emergency situations and the system has the ability to rescind them when the emergency mode is eliminated or the duration expires; (4) Non-Repudiation: Malicious use of privileges under emergency situations are restricted and limited to a limited duration; (5) Concurrency: Multiple emergencies can be handled at one time and the grouped emergencies can be dealt with concurrently. We contend that the first three evaluation metrics are necessary conditions for AC4E's adaptiveness, while the last two metrics are necessary conditions for AC4E's proactivity.

4.3 Primitives

There are several principal assumptions with respect to the operations. (1) The AC4E model is assumed to be deployed in mission-critical CPSs. (2) The administrator who manages the AC4E model can be trusted, and the role sets, privileges, ACLs and etc., are available after the AC4E model is setup by the administrator. (3) As in [30], the AC4E model assumes that the underlying environment has an authentication system which can indentify each subject reliably. (4) Through the techniques in [31], we also assume that all the emergencies can be detected reliably, and their properties and types are known accurately during the detection period. (5) Although each subject may have many roles, they can only activate one at one time. (6) Finally, all the components of AC4E model are time-synchronized and the CGP is reliable

to provide good data from the cyber and physical worlds just like the assumptions in any other existing RBAC like access control models.

Table 1. Sets and Tables in AC4E model

Name	Definition	Type
S	Entities who use services in the system, $S = set\ of\{< Sid, r >\}, where\ Sid = unique < string >, r \in R$	Set
R	Responsibilities of the corresponding subjects, $R = set\ of\{< role >\}$	Set
O	Entities who provide services in the system, $O = set\ of\{< Oid, ACLs >\}, where\ Oid = unique < string >$	Set
AE	Active emergencies occurring in the current system state, $AE = set\ of\{< Eid, E_{ed} >\}, where\ Eid = unique < string >$	Set
PR	Authorizations of entities in the system, $PR = set\ of\{< options: read, write, execute >\}$	Set
ACL	Access control list which includes roles and privileges of an object, $ACL = set\ of\{< r, pr >\}, where\ r \in R \wedge pr \in PR$	Set
ORT	Tables which store original roles of subjects for recovery options towards normal state, $ORT = set\ of\{< s, r >\}, where\ s \in S \wedge r \in R$	Table
SRT	Tables which store the pair of active roles and the subject, $SRT = set\ of\{< s, r >\}, where\ s \in S \wedge r \in (R \cup er\_role)$	Table
ST	The table which stores the available subjects in the system	Table
S_Subject	A dynamic list of selected subjects to perform responsive actions	List
G	Dynamic group list of emergencies needed to be handled, $G = set\ of\{< g_1, g_2, \cdots, g_n >\}, where\ n < N$	Table

In the AC4E model, the constituent entities maintained in each sensor/actuator node are divided into two groups, objects and subjects. And the space roles in the model are also divided into two types, normal roles and emergency roles. According to the real situation where the CPSs are deployed, the AC4E model adjusts the roles of the corresponding subjects dynamically and proactively in order to execute the responsive actions. Access to the resources or services is controlled by the *Access Control Lists* (*ACLs*) which are maintained by the system. The *ACLs* maintain all the space roles the subjects can be assigned and the corresponding privileges. Let *PE* be the set of physical events in the surrounding environment and $EE \subset PE$ be the set of emergency events which are involved in specific CPSs. *AE* denotes the set of all the active emergencies. AE_i denotes the active emergencies when event *i* occurs, $i \in EE$. And E_i is used to evaluate the effect of event *i* on the system. More details and other basic principal components are listed in **Table. 1**.

4.4 Detection and Preparation

This phase is responsible for the preparation work of the emergency management processes. The types of emergencies are identified immediately using the EII component in the detection stage, and properties and effects are also detected in the preparation stage. We model the CPSs as linear control systems, which are equipped with a Kalman filter and a status detector. We assume a CPS has Linear Time Invariant (LTI) dynamics, described by the following form:

$$\begin{cases} x_{t+1} = A \cdot x_t + B \cdot u_t + C \cdot \omega_t \\ x_0 = x(t=0) \end{cases} \tag{2}$$

where $x_t \in R^n$ is the state vector of the physical environment at time *t*, $u_k \in R^p$ is the control input from the corresponding CPS, $x_t \in R^n$ is the process noise at time *t*, and x_0 is the initial state of the physical system. Note that, in this section, x_0 and ω_k are independent Gaussian random values, and $\chi_0 \in N(0,\Sigma), \omega_k \in N(0,Q)$

A wireless sensor and actuator network is deployed to monitor the physical environment, collect sensitive data and assist to execute the feedback control. All the data are sent to the centralized system status estimator to detect whether the system status has become emergency status. The observation estimator can be described as:

$$y_t = L \cdot x_t + \varpi_t \tag{3}$$

where $y_t = [y_{t,1}, y_{t,2}, \cdots, y_{t,m}]^T \in R^m$ is the measurement vector from sensors/actuators deployed in the physical environment, *m* is the number of sensors and actuators, $y_{t,i}$ denotes the measurement made by node *i* at time t, and $w_k \in N(0,Q)$ is the measurement noise independent of x_0 and ω_k.

A Kalman filter is employed to compute the state estimation $E(x_t)$ from the corresponding observation y_t:

$$\begin{cases} E(x_{t+1}) = A \cdot E(x_{t+1}) + B \cdot u_t + O \cdot [y_{t+1} - K \cdot (E(x_{t+1}) + B \cdot u_t)] \\ K = PC^t \cdot [CPC^t + R]^{-1} \\ P = \lim_{t \to \infty} P_{t|t-1} \end{cases} \tag{4}$$

And the estimation deviation can be computed by:

$$D(x_t) = x_t - E(x_t) \tag{5}$$

Therefore, the system state detection function can be written as:

$$S\_D_t = \begin{cases} 1, if\ (D(x_t) \le Th) \\ 0, otherwise \end{cases} \tag{6}$$

where *Th* denotes the largest tolerance threshold value, *0* and *1* denotes that the system in emergency state and normal state, respectively.

In order to guarantee the responsiveness to emergencies, timely detection should be provided. We employ a daemon to monitor the system state in a time interval Δt which is small enough to guarantee quick detection of all the emergencies. And this time value can be evaluated by:

$$0 \le \Delta t \le \omega \cdot \min\{e_0, e_1, \cdots, e_n\}. \tag{7}$$

where $\omega (0 \le \omega \le 1)$ denotes the slow factor for emergency detection. A larger ω can make the emergency detected more quickly, while the demand for novel parallel control policies and high performance platforms will increase.

4.5 Action Generation

The Action Generation Model is used to determine the appropriate responsive actions for the corresponding emergencies in CPSs [32][33]. Once the types and properties are identified by AC4E, the Emergency State Transition Graph (ESTG) is generated immediately.

In the ESTG, each emergency state can transit to another emergency state with an average success probability. Each emergency state can also return to a state towards the normal root state with an average success probability. In this section, we refer them as the EL (Emergency Link) and RL (Response Link), respectively. Two emergency states in the ESTG can be described by the emergency pair $< i, j, EL_{i,j}, RL_{i,j} >$ and $< j, i, EL_{j,i}, RL_{j,i} >$. Note that, $EL_{i,j}$ and $EL_{j,i}$ should meet the requirement:

$$\forall i, j \in AE \wedge Depth\_Com(i, j) \neq 0, EL_{i,j} \otimes EL_{j,i} = 1, RL_{i,j} \otimes RL_{j,i} = 1. \quad (8)$$

The details of the generation algorithm are described as follows.

Algorithm 1. Emergency state transition graph generation
Input: Emergencies *AE*;
Output: Emergency transition graph *TG*;
1: Set normal state *N* as the root node and the current node *CN*
2: FOR each emergency $er_k \in AE_i, i \in EE$ DO
3: IF $\exists N_{er_k}$ THEN
4: IF *Depth_Com(CN,* N_{er_k} *)*>0 THEN
5: UPDATE the possibility property for response link RL_{er_k} for edge <*CN*, N_{er_k} >
6: ELSEIF
7: UPDATE the possibility property for emergency link EL_{er_k} for edge <*CN*, N_{er_k} >
8: ENDIF
9: ELSEIF
10: Insert a child node N_{er_k} for emergency er_k
11: IF *Depth_Com(CN,* N_{er_k} *)*>0 THEN
12: ADD the possibility property for response link RL_{er_k} for edge <*CN*, N_{er_k} >
13: ELSEIF
14: ADD the possibility property for emergency link EL_{er_k} for edge <*CN*, N_{er_k} >
15: ENDIF
16: ENDIF
17: RETURN to current system status *CN=Current_Status_Get ( )*
18: END FOR

4.6 Responsive Action Generation

Upon the occurrence of any emergency event in the physical environment, the access control model should detect it immediately. For a specific interval ϖ, we argue that the emergency can be controlled if $\exists l$ (l denotes a route to the Normal Root State) such that:

$$\Delta_0 + \Delta_{de} + \Delta_{ex} + \kappa \leq \varpi \quad (9)$$

where κ is the independent factor, Δ_0, Δ_{de} and Δ_{ex} denote the initial time, processing time and execution time for the coming emergency, respectively.

If we define $U = \frac{\Delta_{de} + \Delta_{ex}}{\varpi}, R = \frac{\Delta_0}{\varpi}$ as the utilization factor and the responsiveness value for controlling the corresponding emergency, then Formula (9) can be rewritten as:

$$0 \le R \le 1-U-\frac{\kappa}{\varpi} \le 1-U \tag{10}$$

Therefore, we can get the responsiveness threshold RT as:

$$RT = 1-U \tag{11}$$

Different emergencies may have cyber dependency, physical dependency or cyber physical interaction dependency. In order to control the system from emergency state to normal state, optimal routes should be selected and evaluated in parallel.

Algorithm 2. Emergency Grouping.

1: FOR all $er_i \in AE$ DO

2: FOR each $e_j \in AE$ DO

3: IF 1== *Dependency_Check(* er_i, e_j *)* THEN

4: $Addlist(g_j, er_i)$

5: ELSE

6: $Addlist(g_i, er_i)$

7: ENDFOR

8: ENDFOR

9: *Create_Threads( )*

10: FOR each emergency group $g_i \in eg$ DO

11: *Orderlist(* g_i, E_{ed}, *Descend)*

12: $Assign\_Thread(T_{id}, g_i, \Delta t)$

13: ENDFOR

14: *Destory_Theads( )*

Algorithm 2 gives the details of the emergency grouping algorithm. The emergencies which have dependencies on each other are assigned to the same group, while other emergencies are assigned to different groups.

As shown in **Fig. 3**, we assume that eight emergencies occur in the CPS concurrently. Before grouping the emergencies, dependencies between different emergencies must be eliminated or controlled first. Physical dependency indicates some of the responsive actions required by different emergencies may need the same subject or environment to execute. It means that different emergencies may have physical conflicts or inclusion relationships. Physical emergency dependencies with inclusion relationships exist among emergency 2, 7, 4, 10 and 5, while emergencies 1, 3 and 8 have physical conflicts. According to Algorithm 2, emergencies 1, 3 and 8 are assigned to Group 1, emergencies 2, 7, 4, 10 and 5 are scheduled to Group 2, and 6 is divided into Group 3.

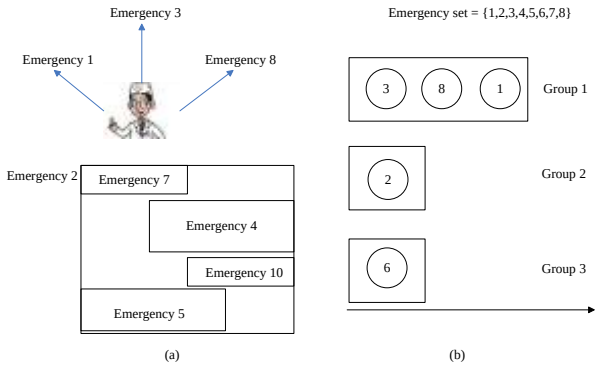


Fig. 3. An example of state transition graph for emergencies

Obviously, in Group 2 emergencies 7, 4, 10 and 5 do not need to be handled any more. Since different emergencies may do the damage to the same CPSs with varying degrees, we use E_{ed} to evaluate the response level required for taking appropriate actions to handle the corresponding emergency events and make the CPSs return to the normal state. When scheduling the emergencies in the same group, window of opportunities, execution time, and successful probability are also considered seriously. All of these statistical data are computed using the data which are involved in the past transactions, including successful and unsuccessful transactions.

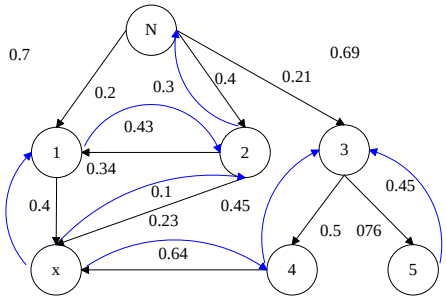


Fig. 4. An example of state transition graph for emergencies

In a specific state *x*, as shown in **Fig. 4**, there are multiple routes from the current state back to the normal state, including $x \to 1 \to N$, $x \to 1 \to 2 \to N$, $x \to 2 \to N$ and $x \to 4 \to 3 \to N$ .The goal is to find the optimal transition route which has the highest probability of success to reach the normal state node. The probability of the route from *x* to *n* can be computed by:

$$\begin{aligned} PS_{x,n} &= (1-p_{x,t}) \cdot p_{x,n} + p_{x,t} \cdot PS_{t,n} \\ &= [p_{x,n}, p_{x,t} \cdot p_{t,n}, p_{x,t} \cdot p_{t,t-1} \cdot p_{t-1,n}, \cdots, p_{x,t} \cdot p_{t,t-1} \cdot p_{t-1,t-2} \cdots p_{1,n}] \\ &\times \begin{pmatrix} 1-p_{x,t} & \cdots & 0 \\ 0 & 1-p_{t,t-1} & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} + p_{x,t} \cdot p_{t,t-1} \cdot p_{t-1,t-2} \cdots p_{1,n} \end{aligned} \tag{12}$$

where $p_{x,t}$ denotes the average ability associated with the arc from state *x* to the immediate state node *t*.

Therefore, the Optimal Route (OR) from the emergency state *x* to the normal state should meet the following requirement:

$$OR = \begin{cases} \max\limits_{\forall RL_i \in Ro} PS_{x,n}, R \geq RT \\ 0, R < RT \end{cases} \tag{13}$$

Note that, not all the best routes which hold the *OR* values are selected to perform responsive actions for the chosen subjects. This is because that the responsive route should meet the principal rule given in Formula (9).

4.7 Execution

Given the identified emergency responsive actions in the responsive actions generation phase, the execution phase identifies the subjects and provides them with the appropriate access privileges to perform responsive actions. Once the actions to be taken at the current system state are identified, the subjects that are selected to perform these actions are chosen. Then the actions need to be enabled and the subjects need to be notified. This can be done in four steps. (1) Provide alternate emergency privileges for the selected subjects. (2) Notify the chosen subjects their new emergency privileges. (3) Rescind the alternate privileges which are activated just for the specific emergency situations after the responses have been performed or the corresponding durations of the emergency privileges are expired. (4) Once the duration waiting timer times out, the system state changes from this kind of emergency situation to another state. All these alternate privileges are provided by the selected subjects with new temporary emergency roles and add new entries to the corresponding objects' *ACL*s. The AC4E informs the selected subjects of their new subjects' roles and objects' *ACL*s after changes are made to the subjects' role and objects' *ACL*s. The system is also responsible for maintaining the detailed records and events.

Algorithm 3 illustrates the main process of AC4E, which gives the specific pseudo codes and can monitor, detect and control the transition of system state. The function *Current_Status_Get( )* is used to check the system state. If a state change is detected, it checks whether the system has now changed from normal mode *N* to emergency mode. The function *Emergency_ID_GET(t)* is used to get the emergency *ID* set in which emergencies are active at time *t*. Then, the subjects that are selected to execute for the corresponding emergencies are collected to *S_Subject*. Once *TS* is known, the selected subjects are provided with the temporary emergency privileges to execute the necessary responsive actions using *AddACL( )* and *Active_role( )* fuctions. Then using *INFORsub( )*, each selected subject is informed the new emergency roles and the corresponding privileges. All actions in the system are recorded by the function *recordActions( )*. Note that, this function is just used to ensure accountability to the AC4E model that any malicious actions or behaviors to the CPSs can be detected correctly and timely. Once the emergency privileges are provided for the selected subjects and roles, the system waits for T_d duration of time, and repeats the whole process again. Note that, the alternate privileges provided for the chosen subjects in the previous state are temporary and will be rescinded once they have taken the required actions or they are no longer required to perform these actions in a new system state. Therefore, the privileges' conflicts which may happen between any roles are avoided.

Algorithm 3. AC4E execution.

1: The selected subject: set $s\_subject = \varnothing$
2: *mode*←*Normal*

3: t←N

3: WHILE (TRUE) DO
4: t: =*Current_Status_Get( )*
5: IF (t==*N*) THEN
6: *mode*←*Normal*
7: FOR each $e \in OldRoleTable$ DO
8: $Active\_role(e.s, s_{ad\min}, e.r)$
9: ENDFOR

10: ELSEIF
11: *mode*←*Emergency*

12: ENDIF
13: IF mode==Emergency THEN
14: FOR all emergency groups ***G*** DO
15: *er: = Emergency_ID_GET(t)*
16: FOR each $ST_i \in ST$ DO
17: IF $er == ST_i$ THEN
18: $s\_subject = s\_subject \cup \{ST_i, x\}$
19: ENDIF

20: ENDFOR

21: FOR each $a_i \in TS$ DO
22: $AddACL(er, ai.p, ai.o, s_{ad\min})$
23: ENDFOR

24: FOR each $s \in s\_subject$ DO
25: $er\_role := SRT(s)$
26: $OldRoleTable := OldRoleTable \bigcup \{(er\_role, CurrentRole \mid s)\}$
27: $Active\_role(s, s_{ad\min}, er\_role)$
28: ENDFOR

29: $INFORsub(s\_subject, TS, er)$
30: ENDFOR

31: *recordActions( )*
32: ENDIF
33: Wait (t_ε)
34: ENDWHILE

5. Validation of AC4E model

In this section, we prove the AC4E model can meet the following properties, responsiveness, correctness, safety, non-repudiation, and concurrency.

Proposition 1. **Responsiveness**: *When an emergency occurs, the system can detect it immediately and notify the selected subjects their changed access privileges*.

Proof. The AC4E model can detect the emergency situations periodically using the function *Current_Status_Get( )* in Algorithm 3 on Lines 4-5. Lines 22, 27 and 29 are used to change the permission associations and to notify the selected subjects. And the emergency role (promotion or demotion) of the subjects are added to the *ACLs* by the function *AddACL( )* on Line 21 and 22.

Proposition 2. **Correctness**: *Subjects get the emergency roles and the alternate privileges if and only if the system is running under emergency situations.*

Proof. If there is at least one uncontrolled emergency in the system, the working mode of AC4E is set to *Emergency Mode* in lines 4-12 of Algorithm 3. This results in the execution of Lines 13-32. Then emergency roles and the corresponding alternate privileges are assigned to the selected subjects on Lines 22 and 27. If a subject is allowed to use the emergency roles and alternate privileges, the system must work under the *Emergency Mode* on line 13. As this can happen only if the return value of *Current_Status_Get* in Line 4 is *Emergency*, the result follows.

Proposition 3. **Safety**: *The temporary roles and privileges can only be used under emergency situations and the system has the ability to rescind them when the emergency mode is eliminated or the duration is expired.*

Proof. The temporary roles and privileges which are activated to cope with the emergencies can be used to selected subjects during the right duration in Line 33. Once the system state moves to the normal state, these roles and privileges cannot be used anymore because of the execution of Lines 7-8.

Proposition 4. **Non-Repudiation**: *The malicious use of privileges under emergency situations are restricted and limited to a finite duration.*

Proof. Line 31 of Algorithm 3 ensures that all the changes related to the responsive actions are recorded in the log files, such as the promotion or demotion of roles, the activation of the temporary privileges and notifications to the subjects. Based on Proposition 2 and 3, it follows that subjects are granted emergency privileges only in the presence of emergencies and the maximum duratizon, and the potential malicious actions are limited to a finite amount of time.

Proposition 5. **Concurrency**: *Multiple emergencies can be handled at one time and the grouped emergencies can be dealt with concurrently.*

Proof. All the active emergencies are detected and each *Emergency_ID* is stored in a group according to the dependency between the remaining emergencies in Lines 1-8 of Algorithm 1. Then the execution of Lines 10-13 results in that each emergency group is managed by a lightweight thread which is created by the main process of Algorithm 2. Therefore, multiple emergency events can be handled and controlled concurrently.

6. Case Study

In this section, a case study of medical CPSs shows the ability of AC4E model to handle multiple active emergencies under emergency situations for CPSs. Consider a medical CPSs in which 7 types of emergencies happened 19 times. All the emergencies occurred in the system are shown in **Fig. 5**.

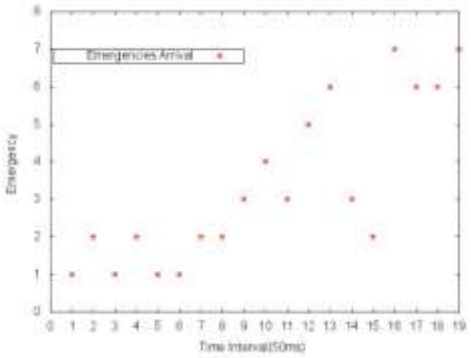


Fig. 5. Emergecies occur in the case study

First, the state transition graph for all the active emergencies of the CPS is generated using Algorithm 1. It can be used to determine the appropriate responsive actions for emergencies in the system. As shown in **Fig. 6**, each emergency pair has two links with the corresponding success probabilities. And each emergency state has at least one route to the root node.

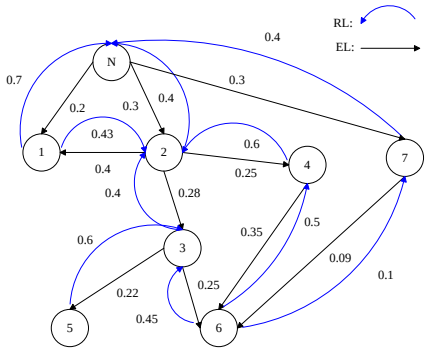


Fig. 6. The state transition graph

The details of each emergency and its corresponding important characteristics are all provided in **Table 2**. During the time interval between 0 ms and 95ms, the arrival sequence and detailed information, such as the subjects, the minimum time window of opportunities, the executing time and the possibility to handle and control the emergency successfully in this transaction are listed in **Table 2**.

Table 2. Properties of Each Emergency

Time	Emergency	Subject	O(mins)	E	True/False
1	E1	U1	2	1.4	T
2	E2	U3	4	+∞	F
3	E1	U1	2	1.5	T
4	E2	U3	4	3.5	T
5	E1	U1	2.5	+∞	F
6	E1	U1	2.0	1.6	T
7	E2	U3	3.5	3.2	T
8	E2	U3	3.5	+∞	F
9	E3	U1,U3	6	3.5	T
10	E4	U4	5	4.5	T

11	E3	U1,U3	6	+∞	F
12	E5	U2,U1	3	3.6	T
13	E6	U3,U2	5	2.7	T
14	E3	U1,U3	5	4	T
15	E2	U3	3.5	2	T
16	E7	U7	5	+∞	F
17	E6	U3,U2	5	+∞	F
18	E6	U3,U2	5	2.9	T
19	E7	U7	5	2.3	T

As shown in **Table 3**, according to **Table 2**, we can compute the *ED* value for each emergency, and then get the corresponding priorities of the CPS system. Obviously, E3, E5 and E6 have physical conflicts, since they employ the same subjects to perform the corresponding responsive actions. And the responsive actions which are necessary for E4 are included in the responsive actions of E2. Therefore, When E4 and E2 occur in a quite small time interval, we only need to deal with E2, and thus E4 is eliminated. Both of the two relationships between different emergencies are Physical Dependencies.

Table 3. Emergency Degree and the Corresponding Priorities

Emergency	ED	Priority Level
E3	3.38889	1
E5	2.86667	2
E6	2.82233	3
E7	2.76667	4
E4	2.62500	5
E2	2.33333	6
E1	1.41667	7

According to Algorithm 2, the active emergencies are divided into four different emergency groups. The dependency between different emergencies is eliminated during the execution of Algorithm 2. Then the four groups are handled by 4 threads generated by the main process of Algorithm 2, respectively. The available responsive routes for each emergency are selected from the emergency state transition graph in **Fig. 6**. As shown in **Fig. 7**, these emergency responsive routes are handled by AC4E in four parallel groups. When the system is under state E6 or E1, there are two alternate responsive routes towards the normal state.

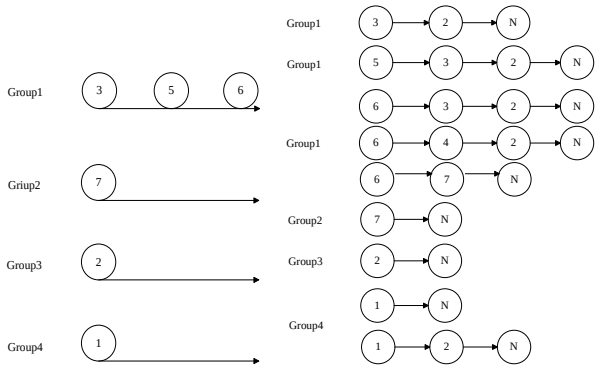


Fig. 7. The Emgency Groups and the Alternate Routes for Each Emergency

After the available routes are chosen from the transition graph for emergencies, AC4E selects the optimal responsive route for each emergency from all the alternative routes, as shown in **Fig. 7**. Consider a CPS system under two situations.

Case 1: Assume $\varpi = 10, \Delta_0 = 1, \Delta_{de} = 0$, then each emergency responsive route can be evaluated by

Group 1:

E3: R=0.1, RT=0.8, $P_{2,n}$=0.4,

E5: R=0.1, RT=0.7, $P_{3,n}$=0.16,

E6: R=0.1, RT=0.7, $P_{3,n}$=0.16, $P_{4,n}$=0.24, R=0.1, RT=0.8, $P_{7,n}$=0.022

Group 2:

E7: R=0.1, RT=0.9, P=0.4,

Group 3:

E2: R=0.1, RT=0.9, P=0.4,

Group 4:

E1: R=0.1, RT=0.9, $P_{1,n}$=0.4, R=0.1, RT=0.8, $P_{2,n}$=0.564

For Case 1, each responsive route can reach the normal state through several hops. For emergency E6, the optimal route is {<6, 4, 2, N>}, while the route {<1, 2, N>} is better for E1 than its other responsive routes. All the routes are selected if and only if they meet the requirement and basic constraints illustrated in Formula 9.

Case 2: $\varpi = 3, \Delta_0 = 1, \Delta_{de} = 0$, then each emergency responsive route can be evaluated by

Group 1:

E3: R=0.333, RT=0.333, $P_{2,n}$=0.4,

E5: R=0.333, RT=0, $P_{3,n}$=0

E6: R=0.333, RT=0, $P_{3,n}$=0, $P_{4,n}$=0, R=0.333, RT=0.333, $P_{7,n}$=0.022

Group 2:

E7: R=0.333, RT=0.667, P=0.4,

Group 3:

E2: R=0.333, RT=0.667, P=0.4,

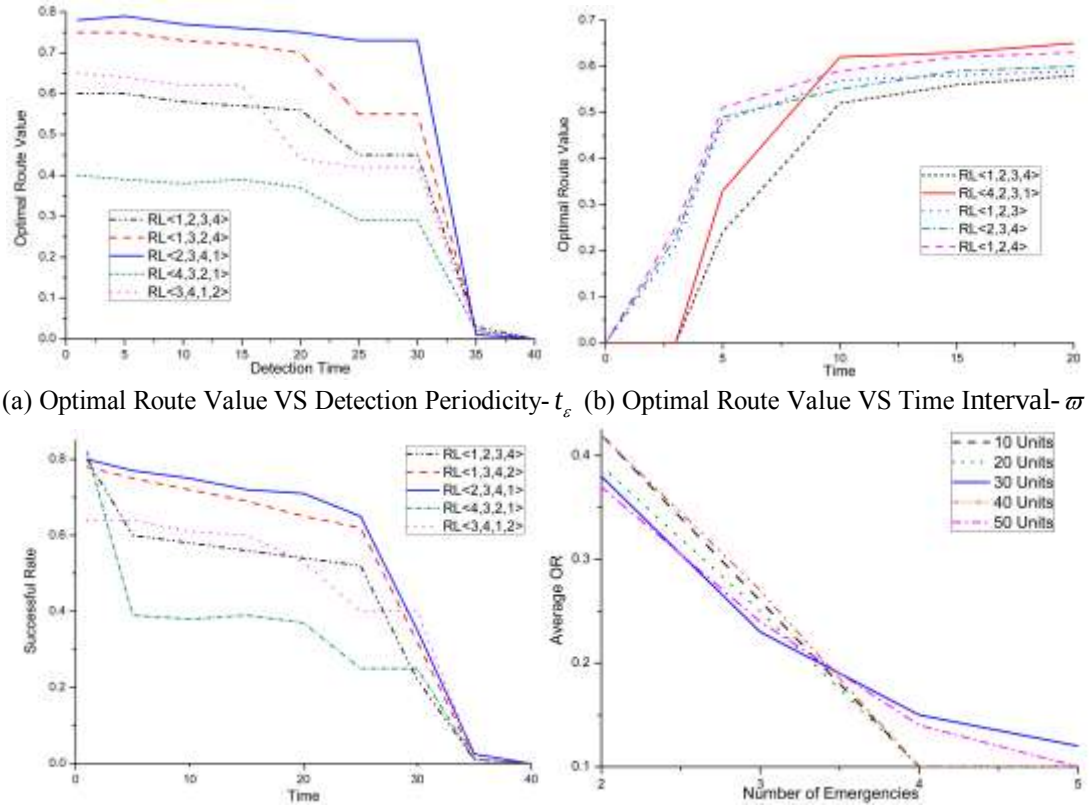
Group 4:

E1: R=0.333, RT=0.667, $P_{1,n}$=0.4, R=0.333, RT=0.333, $P_{2,n}$=0.564

For Case 2, during the period of 0 and $\varpi = 3$, emergencies E3, E6, E7, E2 and E1 have the available responsive routes to the root. For emergency E6, AC4E can provide only one available route, {<6, 7, N>} based on the evaluation of the whole situation, while all the routes for E5 are unavailable, since they do not meet the basic constraints given in Formula 9.

7. Simulations and Discussion

In this section, we present simulation based study to understand the performance and behaviors of the AC4E access control model. For the sake of simplicity, we present simulations for a CPS system in which 4 different emergencies are active at the current time. Obviously, the simplified CPS system can be easily extended to manage more emergencies.



(a) Optimal Route Value VS Detection Periodicity- t_ε (b) Optimal Route Value VS Time Interval- ϖ

(c) Successful Rate VS Detection Periodicity- t_ε (d) Average OR Value VS Emergencies Number

Fig. 8. Simulation Studies

As shown in **Fig. 8**, we have finished four groups of simulations. **Fig. 8 (a)** is the study result of variation of *OR* value for each emergency state with respect to the periodicity of emergency detection t_ε which is on Line 32 in Algorithm 3. As expected, we find that as the t_ε value increases, the corresponding *OR* value remains steady or decreases during the period from 1 to 30 time units. After 35 units, all the *OR* values decline sharply to 0. The main reason is that we increase the time interval between two emergency detection cycles, and then the responsive actions are delayed after the emergency events have occurred. As introduced in Section 4, the CPSs can take an optimal route towards the normal system state using the AC4E model based on the *OR* value. In this group, the average probabilities associated with the optimal route to reach the normal state from all the emergency states are computed. The second group of simulations are the study of the *OR* values with respect to the time interval ϖ in Formula (9). As shown in **Fig. 8 (b)**, as the ϖ value increases, the corresponding *OR* value increases gradually during the period from 5 units to 20 units. For the interval of 0 units to 5 units, the *OR* values of the emergency state with four members are lower than the emergency state which three members. This is because that the hops towards the normal state of *RL* link of <1, 2, 3> state are smaller than that of <1, 2, 3, 4>, and thus the former can reach the normal system state more easily. **Fig. 8 (c)** studies the variation of the successful rate with respect to the value of t_ε. The trend of different successful rates is similar to that of **Fig. 8 (a)**, since as the t_ε value increases, the route which has the best *OR* value does not meet the requirement of Formula (9) any more. The AC4E model has to select another responsive route with the next

best *OR* values to return to the root. The last group of simulations are the study of the average *OR* values with respect to the number of the active emergencies in the CPS system. As shown in **Fig. 8 (d)**, the number of emergencies varies inversely to the average *OR* values. This is mainly because that as the number of active emergencies increases, the chances of satisfying Formula (9) decrease, and then the probabilities of the *OR* values decline at the same time.

8. Conclusion and Future Work

This paper presents an adaptive access control model called AC4E for emergencies for mission-critical CPSs. It not only provides the ability to control access to data and services in normal situations, but also grants the correct set of access privileges, at the correct time, to the correct set of subjects in emergency situations. It can facilitate responsive actions adaptively altering the privileges to specific subjects in a proactive manner without the need for any explicit access requests. The AC4E model has the following properties, responsiveness, correctness, safety, non-repudiation and concurrency. Semiformal validation and case-study are given which demonstrate that the AC4E model detects, responses, and controls the emergency events for a typical CPS adaptively in a proactive manner. Eventually, a wide set of simulations are presented to validate the effectiveness and correctness of the AC4E model. However, the performance of the grouping algorithm of AC4E access management model can be improved furthermore. And more simulations should be done under urgent situations.

References

[1] W. Wolf, “Cyber-physical System”, *Computer*, vol.42, no.43, pp.88-89, 2009. Article (CrossRef Link)

[2] R. Poovendran., “Cyber-physical systems close encounters between two parallel worlds”, in *Proc. of the IEEE*, vol.98, no.8, pp.1363-1366, 2010. Article (CrossRef Link)

[3] K. D. Chang and J. L. Chen, “A survey of trust management in WSNs, internet of things and future internet”, *KSII Transactions on Internet and Information Systems*, vol.6, no.1, pp. 5-19, January, 2012. Article (CrossRef Link)

[4] C. Z. Lai, H. Li, Y. Y. Zhang and J. Cao, “Security Issues on Machine to Machine Communications”, *KSII Transactions on Internet and Information Systems*, vol.6, no.2, pp.498-514, Feb.2012. Article (CrossRef Link)

[5] A.Banerjee,K. K. Venkatasubramanian, T. Mukherjee and S. K. S. Gupta, “ensuring safety, security, and sustainability of mission-critical cyber-physical systems”, in *Proc. of the IEEE*, vol.100, no.1, pp.283-299, 2012. Article (CrossRef Link)

[6] L. Insup, O. Sokolsky, et al., “challenges and research directions in medical cyber-physical systems”, in *Proc. of the IEEE*, vol.100, no.1, pp.75-90, 2012. Article (CrossRef Link)

[7] S. Sridhar, A. Hahn, M. Govindarasu, “Cyber-physical system security for the electric power grid”,in *Proc. of the IEEE*, vol.100, no.1, pp.210-224, 2012. Article (CrossRef Link)

[8] M. Yili, T. H. J. Kim, et al., “Cyber-physical security of a smart grid infrastructure”, in *Proc. of the IEEE*, vol.100, no.1, pp.195-209, 2012. Article (CrossRef Link)

[9] J. Sztipanovits, X. Koutsoukos, et al., “Toward a science of cyber-physical system integration”, in *Proc. of the IEEE*, vol.100, no.1, pp.29-44, 2012. Article (CrossRef Link)

[10] M. Chen, S. Gonzalez, V. Leung, Q. Zhang and M. Li,“2G-RFID based E-healthcare System”, *IEEE Wireless Communications Magazine*, vol.17, no.1, pp.37-43, Feb.2010. Article (CrossRef Link)

[11] M. Chen, S. Gonzalez, Q. Zhang and V. Leung, "Code-Centric RFID System Based on Software Agent Intelligence", IEEE Intelligent Systems, vol.25, no.2, pp.12-19, Mar.2010. Article (CrossRef Link)

[12] R. S. Sandhu, E. J. Coyne, et al., "Role-Based Access Control Models", *Computer*, vol.29, no.2, pp.38-47, Feb.1996. Article (CrossRef Link)
[13] S. Chakraborty, I. Ray, "TrustBAC-Integrating trust relationships into the RBAC model for access control in open systems", in *Proc. of the 11th ACM Symp. on Access Control Models And Technologies*, New York: ACM Press, pp.49-58, 2006. Article (CrossRef Link)
[14] D. Povey, "Optimistic Security: A New Access Control Paradigm", in *Proc. of New Security Paradigms Workshop 1999*, pp.40-45, 1999. Article (CrossRef Link)
[15] A. Corrad, R. Montanari and D. Tibaldi, "Context-based access control management in ubiquitous environments", in *Proc. of Third IEEE International Symposium on Network Computing and Applications* , pp.253-260, 30 Aug.2004. Article (CrossRef Link)
[16] K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta , " Caac-An adaptive and proactive access control approach for emergencies for smart infrastructures", *ACM Trans. Autonom. Adaptive Syst. (Special Issue on Adaptive Security)*, to be published. Article (CrossRef Link)
[17] I. Ray and M. Toahchoodee, "A Spatio-Temporal Access Control Model Supporting Delegation for Pervasive Computing Applications", in *Proc. of the 5th International Conference on Trust, Privacy and Security in Digital Business*, pp.48-58, Sep.2008. Article (CrossRef Link)
[18] I. Ray and M. Toahchoodee. "A Spatio-Temporal Role-Based Access Control Model", in *Proc. of the 21stAnnual IFIP TC-11 WG 11.3 Working Conference on Data and Applications Security*, pp.211-226, Jul.2007. Article (CrossRef Link)
[19] S. Yu, K. Ren and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," in *Proc. of IEEE INFOCOM 2009*, pp.963-971, 2009. Article (CrossRef Link)
[20] S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks", *Journal of Computer Communications of Elsevier*, vol.34, no.3, pp.281-294 2010. Article (CrossRef Link)
[21] S. K. S. Gupta, T. Mukherjee and K. Venkatasubramanian, "Criticality Aware Access Control Model for Pervasive Applications", in *Proc. of the 4th IEEE Conference on Pervasive Computing and Communications*, pp.251-257, 2006. Article (CrossRef Link)
[22] K. G. Christos and M. Ioannis, "Flexible Team-Based Access Control Using Contexts", in *Proc. of the sixth ACM symposium on Access control models and technologies*, ACM SIGSAC, pp.21-27, 2001. Article (CrossRef Link)
[23] G. W. Wu, D. Z. Lu, et al., "A fault-tolerant emergency-aware access controls scheme for cyber-physical systems", *Information Technology and control*, vol.40, no.1, pp. 29-39, 2011. Article (CrossRef Link)
[24] S. Yu, K. Ren, W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," in *Proc. of IEEE INFOCOM 2009*, pp.963-971, 2009. Article (CrossRef Link)
[25] L. Sha et al., "Cyber-physical systems: A new frontier," *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*, 2009. Article (CrossRef Link)
[26] O. D. Mohatar, A. F. Sabater, J. M. Sierra, "A lightweight authentication scheme for wireless sensor networks", Ad Hoc Networks, vol.9, no.5, pp.727-735, Jul.2011. Article (CrossRef Link)
[27] N. Li, N. Zhang, S. Das, et al., "Privacy preservation in wireless sensor networks: A state-of-the-art survey," *Ad Hoc Networks*, vol.7, no.8, pp.1501-1514, Nov.2009. Article (CrossRef Link)
[28] B. Carbunar, Y. Yu, W. Shi, et al., "Query privacy in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol.6, no.2, pp.1-34, 2010. Article (CrossRef Link)
[29] R. Zhang, Y. Zhang and K. Ren, "DP2AC: Distributed privacy-preserving access control in sensor networks", in *Proc. of INFOCOM 2009*, pp.1251-1259, 2009. Article (CrossRef Link)
[30] T. Mukherjee, K. Venkatasubramanian, et al., "Performance Modeling of Critical Event Management for Ubiquitous Computing Applications", in *Proc. of The International Symposium on Modeling, Analysis and Simulations of Wireless and Mobile Systems*, ACM/IEEE, pp.12-19, 2006. Article (CrossRef Link)
[31] M. Sloman and E. Lupu, "Security and Management Policy Specification", *IEEE Network*, vol.16, no.2, pp.10-19, Apr.2002. Article (CrossRef Link)

[32] J. F. Wan, H. H. Yan, H. Suo, et al., "Advances in cyber-physical systems research", *KSII Transactions on Internet and Information Systems*, nol.5, no.11, pp.1891-1908, Nov.2011. Article (CrossRef Link)
[33] D. Chen and G. R. Chang, "A survey on security issues of m2m communications in cyber-physical systems", *KSII Transactions on Internet and Information Systems*, vol.6, no.1, pp.24 - 45, Jan.2012. Article (CrossRef Link)

Dong Chen received his master degree in Computer Science in July 2010 at Northeastern University, China. Now, he is a Ph.D. candidate in Computer Science at Northeastern University and University of Massachusetts Amherst, U. S. His main research interests are Security of Cyber-Physical Systems and Internet of Things.

Guiran Chang is a Professor at Northeastern University, China. He received his bachelor degree in 1970 from Tsinghua University and his Ph.D. degree in 1991 from the University of Tennessee, U. S. His research interests include: P2P, Wireless Sensor Network, Cyber-Physical Systems, and Cloud Computing.

Jie Jia is currently an Associate Professor in the college of Information Science and Engineering, Northeastern University. She received the PhD degree in computer science from Northeastern University, Shenyang, China, in January 2009. Her research interests include wireless networks, mobile Internet, and cognitive radio technology, etc.