Modeling Access Control Resource Based on Process Algebra

Wang Lisong Qin Xiaolin Ding Qiulin

College of Information Science and technology, Nanjing University of Aeronautics and Astronautics,
29 Yudao Street, Nanjing, 210016,P. R. China

Summary
In this paper, the rights, security labels, roles, etc are called access control resources. A calculus with resource usage and consumption is proposed in this paper to model access control resource, it is a new variation of the CCS, named calculus of resource usage and consumption (RUCC for short), in which a process must have and consume some resources to execute an action. In RUCC, processes operate relative to a resource environment, and communications can only happen if principals have provided sufficient resources for the communication action. In this preliminary paper, we design the syntax, semantics for the calculus and some examples show that this calculus has very powerful expressiveness.

Key words:
process algebra, access control resources, security policy

1. Introduction

In computer system, with security requests, such as access control policy[2,3,4,5,6], the executing of a process always are restricted. For example, a process must have the authorization or security level to perform accessing an object, or it must be a certain role, in a range of IP address if it wants to access an object. A process access an object can be interpreted to a process communicates with another process. Sometime this communication may consume some resource. For example a user want to download a music file from a web site, he must click 10 times advertisement and pay for 5$. We use *resource* to describe all of the things that include rights, security labels, roles, or something which process can consume for perform an action. Using the notion of *resource*, we can say that a process must hold certain *resources* or consume some *resource*s during communicating with another process. We assume that there exists a resource set $\Re$, $r \in \Re$, and $num : \Re \to$, father, there exists a binary relation between two resources, called domination, noted as $\pm$, and it is a partial order.

In this paper, we propose a process calculus for process must be equipped enough resources to execute action. The calculus is named RUCC, from syntax aspect, it is a variation of the CCS[1]. We will give the definition of syntax, semantics and equivalence for RUCC, and some example can illustrate that RUCC is very useful, such as security polices specification.

Manuscript received March 5, 2010
Manuscript revised March 20, 2010

2. The language

RUCC is an extension of Milner's CCS[1], where process is relative to a resource environment and the prefix action is banded a condition. So the base part of RUCC syntax is based on the same elements as CCS. A is the set of name, $\overline{A} = \{\overline{a} : a \in A\}$ is the set of co-name, $L = A \cup \overline{A}$ is the label set, and let P be the set of RUCC processes, ranged over by E, F, P *and* Q. The syntax of RUCC process is defined as follows:

$$P ::= 0 \mid b \to a.P \mid P_1 + P_2 \mid P_1 \mid P_2 \mid !P \mid P \backslash L \mid A$$
$$b ::= num(r) \geq n \mid b \wedge b \mid b \vee b$$

Where a ranges over the action set: $Act = L \cup \{\tau\}$, $L \subseteq L$, and f is a rename function as in CCS. b is Boolean express, the meaning of $b \to a.P$ is that process $b \to a.P$ can execute action a while b is true. We use P to denote the set of all processes

Suppose a identifier set denoted as ID, every process have a unique identifier. Processes may be "changed" by involving, but their *id* is always no changed. *id* has two properties, one is used to *depict* the behaviour of process, and another is used to express the resources states of system. In this calculus, *id* can be composited, defined as follow:

$$ID ::= id \mid ID_1 + ID_2 \mid ID_1 \mid ID_2$$
Let $IDSet(ID) \square \{id \mid id\ occurs\ \text{in}\ ID\}$

Definition 1. An atomic identifier is the *id* which has no containing any other *id* .

Definition 2 well construction *id* is defined as

(1) An atomic *id* *is* a well construction

(2) If ID_1 and ID_2 are well construction *and* $IDSet(ID_1) \cap IDSet(ID_1) = \varnothing$, *then* $ID_1 \mid ID_2$ and $ID_1 + ID_2$ *are* well construction.

3. Operational Semantics and Equivalences

Let Γ be the environment. An environment Γ is defined as a triple $\langle \Gamma_{pr}, \Gamma_{pid}, \Gamma_{acr} \rangle$ where

(1) $\Gamma_{pr} : /D \times \Re \to$ □ , $\Gamma_{pr}(P,r)$ records the resource states of process.

(2) $\Gamma_{pid} : /D \to Proc$, $\Gamma_{pid}(P)$ records the id of process.

(4) $\Gamma_{acr} : Act \times \Re \to$ □ , $\Gamma_{ar}(a,r)$ records the consumption of action executed. A same action a in distinct id or process will use and consume the same resources.

The pair ($\langle \Gamma, id \rangle$) is called system, and S denotes the set of systems. The meaning of it is that process P has a resource environment Γ , and Γ records the resources states and resource usage of the process.

Definition 3. In an resource environment Γ , $\forall id \in /D$, we have

(1) $\langle \Gamma, id \rangle$ $num(r) \geq n$
iff $\Gamma_{pr}(id,r) \geq n$ *or for some* $r' \in \Re$
such that $(\Gamma_{pr}(id,r') \geq n$ *and* $r' \sqcup r)$

(2) $\langle \Gamma, id \rangle$ $b_1 \wedge b_2$ *iff* $\langle \Gamma, id \rangle$ b_1 *and* $\langle \Gamma, id \rangle$ b_2

(3) $\langle \Gamma, id \rangle$ $b_1 \vee b_2$ *iff* $\langle \Gamma, id \rangle$ b_1 *or* $\langle \Gamma, id \rangle$ b_2

We use the labeled transition system (LTS for short) to defined the operational semantics of RUCC. Firstly we will define the LTS for process without environment. and the transition relation $\to \subseteq P \times Act \times P$ is defined as in the table 1. and then the definition of LTS for the environments and systems are proposed in definition 3 and table 2 respectively.

Table 1. The operational rules of processes for RUCC: $(P,\ Act,\ \to)$

(Prefix) $\dfrac{-}{b \to a.P \xrightarrow{b \to a}_P P}$

(Sum) $\dfrac{P \xrightarrow{b \to a} P'}{P+Q \xrightarrow{b \to a} P'}$ $\dfrac{Q \xrightarrow{b \to a} Q'}{P+Q \xrightarrow{b \to a} Q'}$

(Com_L) $\dfrac{P \xrightarrow{b \to a} P'}{P|Q \xrightarrow{b \to a} P'|Q}$

(Com_R) $\dfrac{Q \xrightarrow{b \to a} Q'}{P|Q \xrightarrow{b \to a} P|Q'}$

($Repl$) $\dfrac{P \xrightarrow{b \to a} P'}{!P \xrightarrow{b \to a} P'|!P}$

(Com) $\dfrac{P \xrightarrow{b_1 \to l} P' \quad Q \xrightarrow{b_2 \to \bar{l}} Q'}{P|Q \xrightarrow{b_1 \wedge b_2 \to \tau} P'|Q'}$

(Restriction) $\dfrac{P \xrightarrow{b \to a} P'}{P \setminus L \xrightarrow{b \to a} P' \setminus L}$ $a \notin L \cup \bar{L}$

(Rename) $\dfrac{P \xrightarrow{b \to a} P'}{P[f] \xrightarrow{f(b \to a)} P'[f]}$

(Constant) $\dfrac{P \xrightarrow{b \to a} P'}{A \xrightarrow{b \to a} P'}$ where A □ P

Table 2. The operational rules for RUCC: $(S,\ Act,\ \to_S)$

($Action$) $\dfrac{\Gamma_{pid}(id) \xrightarrow{b \to a} \Gamma'_{pid}(id), \Gamma_{pr}(id,r) \geq \Gamma_{acr}(a,r) \text{ for all } r \in \Re}{\langle \Gamma, id \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma', id \rangle}$,
$where \Gamma'_{pr}(id,r) = \Gamma_{pr}(id,r) - \Gamma_{acr}(a,r)$ *for all* $r \in \Re$

(Sum_L) $\dfrac{\langle \Gamma, ID_1 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma', ID_1 \rangle}{\langle \Gamma, ID_1 + ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma', ID_1 + ID_2 \rangle}$ $where\ id \in IDSet(ID_1)$

(Sum_R) $\dfrac{\langle \Gamma, ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma', ID_2 \rangle}{\langle \Gamma, ID_1 + ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma', ID_1 + ID_2 \rangle}$ $where\ id \in IDSet(ID_2)$

(Com_L) $\dfrac{\langle \Gamma, ID_1 \rangle \xrightarrow{(id_1,b_1,a)}_S \langle \Gamma^1, ID_1 \rangle}{\langle \Gamma, ID_1 \mid ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma^1, ID_1 \mid ID_2 \rangle}$
$where\ id_1 \in IDSet(ID_1)$

(Com_R) $\dfrac{\langle \Gamma, ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma^1, ID_2 \rangle}{\langle \Gamma, ID_1 \mid ID_2 \rangle \xrightarrow{(id,b,a)}_S \langle \Gamma^1, ID_1 \mid ID_2 \rangle}$
$where\ id \in IDSet(ID_2)$

(Com) $\dfrac{\langle \Gamma, ID_1 \rangle \xrightarrow{(id_1,b_1,a)}_S \langle \Gamma^1, ID_1 \rangle, \langle \Gamma, ID_2 \rangle \xrightarrow{(id_2,b_2,\bar{a})}_S \langle \Gamma^2, ID_2 \rangle \quad \langle \Gamma, id_1 \rangle \ \ b_2, \langle \Gamma, id_2 \rangle \ \ b_1}{\langle \Gamma, ID_1 \mid ID_2 \rangle \xrightarrow{\tau}_S \langle \Gamma^3, ID_1 \mid ID_2 \rangle}$
$where\ id_1 \in IDSet(ID_1), id_2 \in IDSet(ID_2)$

$$\Gamma^3_{pr}(id,r) = \begin{cases} \Gamma^1_{pr}(id,r) \ if\ \ id \in IDSet(ID_1) \\ \Gamma^2_{pr}(id,r) \ if\ \ id \in IDSet(ID_2) \end{cases}$$

Now we introduce the behavior theory based on bisimulation for RUCC.

Definition 4. (Bisimulation) A relation $R \subseteq P \times P$ is a bisimulation if $(E,F) \in R$ implies, for all $a \in Act$,

(1)
$if\ E \xrightarrow{b \to a} E'$

then there exists F', b' *and* $b \to b'$ *such that*

$F \xrightarrow{b \to a} F'$ *and* $(E', F') \in R$

(2)

if $F \xrightarrow{b \to a} F'$

then there exists E', b' *and* $b \to b'$ *such that*

$E \xrightarrow{b \to a} E'$ *and* $(E', F') \in R$

If there exists a bisimulation containing RUCC processes pair (E, F), $E, F \in P$, then E, F are bisimulation equivalent, notation $E \sim F$.

Definition 5. (Bisimulation for systems) A relation $R \subseteq S \times S$ is a bisimulation if $(\langle \Gamma, ID_1 \rangle, \langle \Gamma, ID_2 \rangle) \in R$ implies, for all $a \in Act$,

(1)

if $\langle \Gamma, ID_1 \rangle \xrightarrow{(id, b, a)}_S \langle \Gamma', ID_1 \rangle$

then there exists id_2, b' *and* $b \to b'$ *and* $\Gamma_{pr}(id_1, r) \le \Gamma_{pr}(id_2, r)$ *for all* $r \in \Re$ *such that*

$\langle \Gamma, ID_2 \rangle \xrightarrow{(id_2, b', a)}_S \langle \Gamma', ID_2 \rangle$

and $(\langle \Gamma', ID_1 \rangle, \langle \Gamma', ID_2 \rangle) \in R$

(2) vice versa

If there exists a bisimulation system pair $(\langle \Gamma, ID_1 \rangle, \langle \Gamma, ID_2 \rangle) \in R$, then $\langle \Gamma, ID_1 \rangle, \langle \Gamma, ID_2 \rangle$ are bisimulation equivalent, notation $\langle \Gamma, ID_1 \rangle \sim_S \langle \Gamma, ID_2 \rangle$.

Theorem 1 Suppose $\Gamma_{pr}(id_1, r) = \Gamma_{pr}(id_2, r)$ *for all* $r \in \Re$ and $S_1 \in (\Gamma, id_1)$, $S_2 \in (\Gamma, id_2)$, then $S_1 \sim_S S_2$ *iff* $\Gamma_{pid}(id_1) \sim \Gamma_{pid}(id_2)$

Proof. (outline)

(1) $\Rightarrow$, Let $S_1 \sim_S S_2$, set $R = \{(\Gamma_{pid}(id_i), \Gamma_{pid}(id_j)) \mid \langle \Gamma, id_i \rangle \sim_S \langle \Gamma, id_j \rangle, i, j \in \}$, it is enough to show that $R \in$. Obviously, $(\Gamma_{pid}(id_1), \Gamma_{pid}(id_2)) \in R$. if (1) in definition 5 hold, and $S_1 \xrightarrow{a}_S S_1'$, by $(S, Act, \to_S)$, we have $\Gamma_{pid}(id_1) \xrightarrow{b \to a} \Gamma'_{pid}(id_1)$, in the same way, there exists $\Gamma'_{pid}(id_2)$ such that $\Gamma_{pid}(id_2) \xrightarrow{b \to a} \Gamma'_{pid}(id_2)$, and because $S_1' \sim_S S_2'$, so $(\Gamma'_{pid}(id_1), \Gamma'_{pid}(id_2)) \in R$. On the other hand, if (2) in definition 5 hold, then we also have $(\Gamma'_{pid}(id_1), \Gamma'_{pid}(id_2)) \in R$, so $R \in$ hold.

(1) $\Leftarrow$, Let $\Gamma_{pid}(id_1) \sim \Gamma_{pid}(id_2)$, set $R = \{(S_i, S_j) \mid \Gamma_{pid}(id_i) \sim \Gamma_{pid}(id_j), i, j \in \}$, where $S_i = \langle \Gamma, id_i \rangle$, $S_j = \langle \Gamma, id_j \rangle$ it is enough to show that $R \in \sim_S$. Obviously, $(S_1, S_2) \in R$. when (1) in Definition 4 hold, then from (1) in Definition 4, and in the same environment and same action a in distinct id or process will use and consume the same resources, then (1) Definition 5 hold. Similarly to the condiction (2) in Definition 5, i.e $R \in \sim_S$

Example 1. Someone wants to download a music file from a website, but he must click the advertisement several times, for instance 10 times, and pay 5$. Now, we can use RUCC to express this process.

Let $\Re \{clickednum, fund\}$, $Act \{click, download\} \cup \{\tau\}$, $id \in / D$, $\Gamma_{pr}(id, clickednum) = 0$, $\Gamma_{pr}(id, fun) = 100$, $\Gamma_{acr}(download, fun) = 5$, $\Gamma_{acr}(click, clickednum) = -1$, $\Gamma_{acr}(download, clickednum) = 10$, and $\Gamma_{pid}(id) = !click \mid (num(clickednum) \ge 10) \to download.P$, then download system will involve as follow:

$\langle \Gamma, id \rangle \xrightarrow{click}_S \langle \Gamma^1, id \rangle \xrightarrow{click}_S \langle \Gamma^2, id \rangle \xrightarrow{click}_S \cdots \xrightarrow{click}_S \langle \Gamma^{10}, id \rangle \xrightarrow{download}_S \langle \Gamma^{11}, id \rangle$

After the action *click* executes 10 times, the Boolean expression $num(clickednum) \ge 10$ becomes true, and performs action *download* will consume 5$, and then the user can download the music file. Where for every $\Gamma^i (1 \le i \le 10)$, $\Gamma^i_{pr}(id, clickednum) = \Gamma^{i-1}_{pr}(id, clickednum) + 1$, and $\Gamma^{11}_{pr}(id, fun) = \Gamma^{10}_{pr}(id, fun) - 5$, $\Gamma^{11}_{pr}(id, clickednum) = 0$

In this example, if using the pure CCS to express this process, the expression will become very complex.

$$a.download.P + a.a.download.P + a.a.a.download.P + \cdots + \underbrace{a.a.\ldots a}_{n}.download.P$$

Example 2 (Discretionary Access Control, DAC)

Let $FILE \{ file_i \mid i \in \}$, $Act \{read, write, \overline{read}, \overline{write}\}$, $\Re \{ r_i \mid r_i = (a, file_i), a \in Act, file \in FILE, i \in \}$, $id_1, id_2 \in / D$, $\Gamma_{pr}(id_1, r_i) = 1$ *for some* $r_i \in \Re$, $\Gamma_{pr}(id_2, r_j) = 1$ *for some* $r_j \in \Re$, $\Gamma_{acr}(a, r_i) = 0, \Gamma_{acr}(\bar{a}, r_i) = 0$ *for all* $r_i \in \Re$, and $\Gamma_{pid}(id_1) = a.P, \Gamma_{pid}(id_2) = \bar{a}.Q$, Set $S_1 \langle \Gamma, id_1 \rangle, S_2 \langle \Gamma, id_2 \rangle$, then the rule of DAC can be expressed as follow:

$S_1 \mid S_2 \xrightarrow{\tau}_S S_1' \mid S_2'$

iff $\exists r_i \in \Re$ *such that* $\Gamma_{pr}(id_1, r_i) = 1$ *and* $\Gamma_{pr}(id_2, r_i) = 1$

4. Relative work and conclusion

In this paper, we propose a calculus based on usage and consumption of resources. We show that how a process communicates with another process within an environment by using and consuming resources. We intend this calculus into some directions, such as it can be applied to describe the various access control politics and security properties. There is already some important literature on topics on study about security properties based on process algebra, the representative literature such as [7, 8], Peter Y. A. Ryan study the mathematical models of computer security based on CSP(Communicating Sequential Processes) in [7], Riccardo Focardi and Roberto Gorrieri propose a security process algebra to study non-interference-like properties for computer security in [8]. But all of them, the changeability of security attribute and ability of a process is not mentioned. In this paper, we use the notion of resources to describe the ability which a process access (i.e. communicate with) another process, and the consumption of resources grasp the changing of the access ability. The resources notation is used in [9,10], in [9], David Pym and Chris Toftsa use resources to express memory, CPU, I/O devices etc in computer system, propose a BI logic, and modelling system based on resources process algebra and BI logic. In [10], Matthew Hennessy and Manish Gaur counting the cost in process algebra and propose a costed picalculus, this theory can be applied to the various calculi being developed for web services etc. As we see in this paper, the resource concept is different to the resource concept in literature such as [9] and [10].

In the future we will refine this calculus, and apply it to security properties analysis etc.

References

[1] R. Milner. Communication and Concurrency. Prentice-Hall, 1989.

[2] D.E Bell, L.J LaPadula,. Secure computer systems: Mathematical Foundations. Mitre Corp. Report No. MTR-2574, Vol.I, 1973. An electronic reconstruction, by Len LaPadula,November, 1996

[3] D.E Bell, L.J LaPadula. Secure computer systems: A mathematical model. Mitre Corp. Report No. MTR-2574, Vol.II, 1973. An electronic reconstruction, by Len LaPadula,November, 1996

[4] Ravi S. Sandhu. Lattice-Based Access Control Models. Computer Volume 26 , Issue 11 Nov.1993, Pages: 9-19

[5] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC). Volume 4 , Issue 3 (August 2001), Pages: 224 - 274

[6] Jaehong Park, Ravi Sandhu. The $UCON_{ABC}$ Usage Control Model. ACM Transactions and System Security, Vol.7 No.1, February 2004, Pages: 128-174.

[7] Peter Y. A. Ryan. Mathematical Models of Computer Security. Lecture Notes In Computer Science; Vol. 2171 Pages: 1 - 62 2000

[8] Riccardo Focardi and Roberto Gorrieri. Classification of Security Properties(Part I: Information Flow). Lecture Notes in Computer Science Volume 2171/2001 Pages:331-396 2001

[9] David Pym, Chris Toftsa. Systems Modelling via Resources and Processes: Philosophy, Calculus, Semantics, and Logic. Electronic Notes in Theoretical Computer Science, Volume 172, 1 April 2007, Pages 545-587

[10] Matthew Hennessy, Manish Gaur. Counting the Cost in the Picalculus (Extended Abstract).Electronic Notes in Theoretical Computer Science, Volume 229, Issue 3, 22 July 2009, Pages 117-129

Wang Lisong received the B.S. in Anhui Normal University Anhui china in 1992, and received the M.S. degrees in Nanjing University of Aeronautics and Astronautics Nanjing China in 1995, respectively. He was an associate professor in the College of Information Science and technology, Nanjing University of Aeronautics and Astronautics, 29 Yudao Street, Nanjing, Nanjing, China. His current research interests include information security and process algebra.