Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ATPL- A TRUST MANAGEMENT POLICY LANGUAGE BASED ON TEAMA

[1, 2, 3]QI JING, [1]LI CHENG, [1]SHUBIN LIAO, [1]SUKE LI

School of Software and Microelectronics, Peking University, Beijing, P.R.China

[2]Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, P.R.China

[3]Key Laboratory of Network and Software Security Assurance (Peking University), Ministry of Education, P.R.China

ABSTRACT

Ad Hoc network is a multi-hop wireless network, consist of the self-organizing wireless mobile nodes and without fixed infrastructure, but its openness brings “stranger access” problem. Trust management strategy based on trust evaluation can solve such problems. By entrusting trust certificate with trust value, access control policy can be passed to "strangers", and make them able to access the nodes in Ad Hoc network. In order to achieve the trust management based on trust evaluation, this paper proposes policy language for trust management in Ad hoc network. ATPL is defined based on constraint Datalog, realizes the three trust delegation chain of TEAMA by entrusting authorization rules, and completes the calculation of certificate execution and delegation trust value at the same time. This paper defines the syntax of ATPL, discusses the safety of ATPL rules, and proposes the basic safety convention of ATPL and safety statute. By the discussion of ATPL explanatory semantics, this paper solves the legacy security issue, and proves that ATPL rule is safe under various security constraints defined herein.

Keywords: *Trust Management, Trust Evaluation, Constraint Datalog*

1. INTRODUCTION

The network organization structure of Ad Hoc is loose, and the node is more individual than that of WSNs network, such as the laptop temporary conference network, as well as the temporary network consist of the PDA, smart cameras and intelligent node network, etc.. In Ad Hoc network, every node is highly mobile and free to join or exit, the structure of network is not stable, so the traditional access control can’t solve the access problem issue in Ad Hoc network environment. Trust management[6] based on policy provides a good solution for the distributed access control issue among nodes in Ad Hoc network environment. However, during the trust delegation of the traditional trust management system[7], the delegation subject is divided into trusted and untrusted directly. And if it’s judged as trusted, the subject will grant all authorities of credential. Thus does not match the mindset of human. So with fine-grained division of the trust [8], we can express the trust concept closer to human thinking, and grant different authority sets in credential based on different trust degree. On the basis of access control authorization model, TEAMA[9], this paper defines the trust policy language, APTL for application in Ad Hoc network. By delegating the authorization rule set, we realize the inference of trust delegation chain and the calculation of credential trust degree. By analyzing and evaluating the corresponding factors, we can provide more dynamic trust management.

2. RELATED WORK

2.1Research of Trust Management

Blaze et al first proposed the concept of trust management in 1996 [10], and introduced the first trust management system, PolicyMaker. Then they launched another trust management system, KeyNote. PolicyMaker and KeyNote got rid of previous authorization model based on identity information by binding the public key and authorization assertions, and realized distributed authorization by trust delegation. Subsequently, there had been lots of trust management system, such as SPKI/SDSI [11], QCM, SD3, RT, Oasis, Ponder and so on.

RT was a role-based trust management framework, proposed by Ninghui Li et al on the basis of RBAC[12], SDSI and DL, and could support the attribute-based access control. RT was a widely discussed trust management framework, but there was no explicit third-party authorization in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RT, and it might cause some problems to join new structure into language freely to enhance its expression ability.

2.2TEAMA Authorization model

TEAMA is an authorization model proposed by Qi Jing et al in 2011, based on NIST RBAC. TEAMA proposes authorization depth control based on hierarchical model and threshold, and discusses the value range of delegation threshold, with the value calculation model of hierarchical delegation. TEAMA contains basic model, BTM and trust degree extension, TE. BTM includes trust delegation submodel, TDM and object control submodel OCM. The former defines the authoritative source subject role and the authoritative source object authority, while the latter introduces the object role, complementarity the BTM model.

TE contains subject trust degree, object trust degree, authorization threshold and authorization level. Subject trust degree means the delegation and execution trust degree in the delegation credential of subject role and authority, while object trust degree means the threshold granted to the execution request of subject role and authority. TEAMA distinguishes the trust degree of credential delegation and execution, and express the authority and authoritativeness naturally by using delegation depth.

This paper defines the trust management policy language, ATPL, on the basis of TEAMA authorization model and Constraint Datalog(Datalog^C), which applies the network application of Ad Hoc. APTL is a policy language defined on Datalog^C directly, without language converter, and its semantics is clear and concise, with high availability. Assessment and calculation of trust value are associated with the process of the trust delegation, so ATPL defines authorization predicates with trust value built-in, and extends Datalog^C with the trust calculation functions joined.

3. ATPL SYNTAX

This section defines the syntax of the ATPL, and explains the specific predicates and the semantics of the rule simply.

Standard Datalog does not include function items, but in order to calculate the trust value with the trust entrusting, ATPL introduces the trust calculation[13] function set, F_M .And in order to enhance the expression ability and availability, introduces the direct evaluated constraint function, F_C , which makes the ATPL has the environment al perception.

$$F = F_M \cup F_C$$

$$F_M = \left\{ f\left(\vec{x}\right) : T \mid x_i : T, f \in \left\{+, -, \times \div\right\}\right\}$$

$$F_C = \left\{ \begin{array}{l} f\left(\vec{x}\right) : NT \mid xi : NT, \\ f \in \left\{gettime, attenuation, ...\right\} \end{array} \right\}$$

Here T means types, TY={S, SR, P, OP, ORP, Opr, Obj, OR, T, D, O},O means other element type,NT=TY-{T}

The introduction of function items could undermine the security conditions of Datalog rules, leading to the infinite logic inference set of one logic programs. In this paper, we constraint the use of function items in the following definition of predicates and rules, and discuss the descriptive semantic of ATPL rules with function entry in this chapter.

3.1 Predicates

ATPL predicates include authorization predicates Pa and condition predicates Pc, The general form is:

$$P\left(\vec{x}\right), \vec{x} = x_1, x_2, ..., x_n, x_i \in IF(i = 1, 2, ..., n)$$

3.1.1 Authorization predicates

Authorization predicates act on authorization behavior of subject, so it's also called subject authorization predicates. It's divided into basic form and built-in trust form, respectively for the basic trust delegate and trust delegate based on trust evaluation.

The basic form of the subject authorization predicate is based on standard definition of Datalog predicates, and contains no function items. While built-in trust subject authorization predicates carry information about trust value evaluation, and contains calculation function of trust value.

3.1.2 Condition predicates

Beside authorization predicates, this article treat predicates as condition predicates, used to constraint the items in authorization predicates and condition predicates of rules, in order to add environmental perception to rules.

P_{C_C} ,Comparison predicates: in ATPL, arithmetic comparison operator such as >,<,=,<=,>=,≠ is treated as predicates of special form, used to comparetrust value, trust transfer level and other factors.

P_{C_CO} ,Condition predicates based on constraint domain, ATPL is defined based on Datalog^C ,among which the condition predicates based on constraint domain is expressed with P_{C_CO} . Thus can define organized resources, such as the structure of file and folder. However, the discussion of constraint domain in this article is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

limited to single-variable constraint domain, can't satisfy the ATPL's demand. To add the expression ability of ATPL, we don't discuss the computability of arithmetic comparison predicates with constraint domain, and list it separately.

3.2 Rules

3.2.1 General rule form

Definition 1,The general rule form of ATPL:

$$p(\vec{x}_0) \leftarrow p_1(\vec{x}_1), p_2(\vec{x}_2), ..., p_n(\vec{x}_n)$$

$$p \in Pa \cup Pc\_o, pi \in P$$

The left part of the arrow $p(\vec{x})$ is called rule header, the right part is called rule body.

The ATPL rules contain local rules and credential. Local rules mean the rule defined by the subject and stored locally, consist of authorization rule and condition rues. Credential means the authorization policy used for exchange and signed by the host.

3.2.2 Credential

Definition 2, Credential:

$$x.time_{validity}.p(\vec{x}_0) \leftarrow p_1(\vec{x}_1), p_2(\vec{x}_2), ..., p_n(\vec{x}_n)$$

x is the signatory of credential, x=first_item(p), $p \in \{srr, sp\}$,$p_i \in P$, $time_{validity}$ is the available period of credential.

The signatory of credential is same as the first item of credential rule header predicate, which is the main difference between credential and local policy rule set. The legitimacy of the certificate requires verifying the signature of signatory and the effective time. After the verification, add the credential to the consistency checking of temporary rule set, and get rid of signatory and effective time.

3.2.3 Conditional Rule

Definition 3, Conditional rule

$$p(\vec{x}_0) \leftarrow p_1(\vec{x}_1), p_2(\vec{x}_2), ..., p_n(\vec{x}_n)$$

Trust delegation depth control rules belong to conditional rule, and are constraint fact.

Trust commission depth control rules set:

dct(D,D1,T,TH)←T>=TH,D1=-1,D=-1. (1)

dct(D,D1,T,TH)←T>=TH,D1>D,D≠-1 (2)

dct(D,D1,T,TH)←T>=TH,D1=-1,D>0.(3)

The three rule above, correspond to three value model of trust commission depthrespectively.

3.2.4 authorization rule

The main authorization rule of ATPL are basic authorization rule and authorization rule with trust built in. The former is constructed by the basic form of subject authorization predicate, and the condition predicates without trust value information, so the rules contain no trust value information. The latter consist of authorization predicates with trust built in and condition predicates.

- Delegation authorization rule set Φ
  - ✓ Role members judgment rule, to achieve the trust delegation chain $TrustList_r$

$$ismem(X,Y,R) \leftarrow srr(X,Y,X,R).$$

$$ismem(X,Y,R) \leftarrow srr(Z,Y,X,R), ismem(X,Z,R).$$

  - ✓ Subject authority judgment rule, to achieve the trust delegation chain $TrustList_p$ and $TrustList_{rp}$

$$hasp(X,Y,P) \leftarrow ismem(X,Y,R), srp(X,P,R).$$

$$hasp(X,Y,P) \leftarrow sp(X,Y,X,P).$$

$$hasp(X,Y,P) \leftarrow sp(Z,Y,X,P), hasp(X,Z,P).$$

$$hasp(X,Y,P) \leftarrow sp(Z,Y,X,P), ismem(X,Z,R), srp(X,P,R).$$

- Delegation authorization rule set Φ with trust built in
  - ✓ Role members judgment rule

A. $ismem(X,Y,R,W,W1,TH,D) \leftarrow srr(X,Y,X,R,S,T,TH,D), W = S, W1 = T.$

B. $ismem(X,Y,R,W,W1,TH,D) \leftarrow srr(Z,Y,X,R,S,T,TH,D), ismem(X,Z,R,W2,W3,TH,D1), dct(D,D1,W3,TH), W = W3 * S, W1 = W3 * T.$

  - ✓ Subject authority judgment rule sets

C. $hasp(X,Y,P,W,W1,TH,D) \leftarrow ismem(X,Y,R,W,W1,TH,D), srp(X,P,R).$

D. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(X,Y,X,P,S,T,TH,D), W = S, W1 = T.$

E. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(Z,Y,X,P,S,T,TH,D), hasp(X,Z,P,W2,W3,TH,D1), dct(D,D1,W3,TH), W = W3 * S, W1 = W3 * T.$

F. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(Z,Y,X,P,S,T,TH,D), ismem(X,Z,R,W2,W3,TH,D1), srp(X,P,R), dct(D,D1,W3,TH), W = W3 * S, W1 = W3 * T.$

other authorization rule set Ψ

The delegation authorization rule set Φ and Φ_T above are the core of ATPL rules. Besides, ATPL also contains other authorization rule sets:

$$p(\vec{x}_0) \leftarrow p_1(\vec{x}_1), p_2(\vec{x}_2), ..., p_n(\vec{x}_n)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$p \in P_a - \{ismem, hasp\}$,$p_i \in$ P-{ismem, hasp},extended authorization rule set Φ_E

4. ATPL RULE SAFETY

4.1 Problems in ATPLRule Safely

Basic safety convention Datalog rule: if all variables in rules are constrained, then the rule is safe.

The ATPL definition does not fully comply with the definition of Datalog language, and violate the Datalog basic safety convention, so it is necessary to discuss the security of its rules:

1. Introduction of function item

ATPL introduces arithmetic calculation function F_M in rule header predicate, and there exists definition of recursive rules. Meanwhile, ATPL also introduces of the directly evaluated constraint function Fc . Function items may result in unsafe rule set and generate an infinite derivation.

2. ATPL rule definition may violate Datalog basic safety convention

1)Constrained variables definition is limited relationship to rule body predicate. However, the introduction of function item outside the arithmetic comparison predicate, makes the rule body predicates which have the same name with therule header predicates in recursive rules, may become infinite relations.

2) Datalog basic safety convention does not contain directly evaluated constraint function F_C , so we need to think about the safety of rules which contain F_C function item in predicates.

3) According to the ATPL rule definition, the case that rule body are all condition predicates may appear in certificate, conditional rules, and other authorized rules. In Datalog, all condition predicates are arithmetic comparison predicates, so when rule body only contains condition predicates, Datalog basic safety convention is violated.

4.2 ATPLBasic Safety Convention

Based on the basic safety convention of Datalog rule, we discuss that of Datalog rule.

Definition 4, Rule body recursive predicates: predicates in rule body with the same name in rule header,denoted as P_{b_h} .

ATPL predicates contain authorized predicates $P_a(x_{a1}, x_{a2}, ..., x_{an})$, arithmetic comparison predicate $P_{c\_c}(x_{c\_c1}, x_{c\_c2}, ..., x_{c\_cm})$, condition predicates based on constraint domain $P_{c\_co}(x_{c\_co1}, x_{c\_co2}, ..., x_{c\_com})$ and other condition predicates $P_{c\_co}(x_{c\_co1}, x_{c\_co2}, ..., x_{c\_cos})$. We use subscript h and b to show whether the predicate are in rule header or rule body:

$x_{h\_ai} \in IF \wedge x_{b\_ai}, \in I$

$x_{h\_c\_oj} \in I \wedge x_{b\_c\_oj}, \in IF_{NT} \cup I_T$

$x_{b\_c\_ck} \in IF_{NT} \cup I_T$

$x_{b\_c\_col} \in IF_{NT} \cup I_T$

Definition 5,ATPL extended constrains :

Assume

$\Gamma = \left(U_\eta pv(P_{b\_an})\right) \cup \left(U_\theta pv(P_{b\_c\_o\theta})\right)$,

$\left(P_{b\_an\eta} \neq P_h \wedge P_{b\_c\_o\theta} \neq P_h\right)$

then $Varz \in \Gamma$ or

Varz=a or a=Varz（a is a constant）or

Varz=Varw or Varw=Varz （Varw is ATPL ATPL extended constrained variable）

then we call Varz as ATPL extended constrained variable.

Definition 6,basic safety convention of ATPLrules:

$$\forall Varx \in \left(pv(P_{h\_a}) \cup pv(P_{h\_c\_o}) \cup \left(\bigcup_\theta pv(P_{b\_c\_c\theta}) \right) \cup \left(\bigcup_\lambda pv(P_{b\_c\_co\gamma}) \right) \cup \left(\bigcup_\rho pv(P_{b\_h\rho}) \right) \right)$$

When Varx is the extended constrained variables of ATPL, rule set is safe.

4.3 ATPL Rule Safety Problem and solution

The basic ATPL safety convention above is extension of basic Datalog safety convention based on ATPL, and does not change the safety problem of ATPL rules fundamentally. So we discuss the safety of ATPL rule in three fields:

1) Directly evaluation function

Here we convent the direct evaluation constraint function F_C is local function, the computing process is completed within a limited time in the local, and contains no input parameters. So the direct evaluation constraint function in rule body and its output parameters are fixed, with no security problem left. When we introduce F_C in condition predicates based on constraint domain, we won't break the feature of constraint domain.

2) Rule with rule body only contains by condition predicates:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ATPL condition predicates include arithmetic comparison predicates P_{c_c}, condition predicates P_{c_co} based on constraint domain. According to the basic safety convention of ATPL rules, when the rule body is only consist of P_{c_c} and P_{c_co}, the rule may not be safe.

Extended convention 1: Beside the trust delegation depth control rules, the appearance of arithmetic comparison predicate P_{c_c} must comply with ATPL basic safety convention.

Extended convention 2: Delegation depth judgments predicates dct only appears in the header of trust delegation depth control rules and the delegation authorized rules set Φ_T with trust built-in.

With extended convention 1,the rules that need discussion whose rule body contain only condition predicates, only have one form, that's rule body only contains P_{c_co}, beside the trust delegation depth control rule. According to extended convention 1,the safety of trust delegation depth control rules and the delegation authorized rules set Φ_T with trust built-in will be discussed in the ATP L semantics.

3) Recursive rules with rule header contains arithmetic calculation function items

Extended convention 3: Recursive rules with rule header contains arithmetic calculation function item, only appears in built-in trust delegation authorization rule set Φ_T.

So, the safety of this kind of rule will be discussed with built-in trust delegation authorization rule set Φ_T.

5. EXPLANATORY SEMANTICS OF ATPL

ATPL is defined based on Datalog^C, and Dalaog is a First Order Logic Program. The explanatory semantics of FOLP is mainly about "integrity of the grogram" and model-theoretic semantics, proposed by Clark. The latter is more nature and widely accepted. So below we analyze the explanatory semantics of ATPL, based on the model-theoretic semantics of First Order Logic Program.

5.1 Explanatory Semantics of ATPL BasicLogic Program

The basic logic program P_Ω is standard Datalog program, including limited basic set of credentials, , basic authorized agent set of rules, Φ, other basic authorized agent set of rules, ψ_Ω, and conditional rules set CR_Ω. All rules contain no information about trust value, only realize the trust delegation function among subject.

5.2 Explanatory Semantics of ATPL Logic Program , With Trust Built In

Logic program P_T includes C_T, a limited credential set, Φ_T, an authorization rules set with trust built in ψ_T,an other authorized rule set with trust built in, and CR_T, a conditional rules set. According to the trust value information in rules, we can accomplish the intersubjective trust delegation as well as the calculation of trust evaluation. Because logic program P_T extend the trust calculation function set F_M in the recursive rule header predicate of Φ_T, and add the trust delegation depth control predicates to rules, we need to investigate the safety of P_T.

According to the basic safety conventions of ATPL rules, we investigate Φ_T, the delegation authorized rules set with trust built in.

A. $ismem(X,Y,R,W,W1,TH,D) \leftarrow srr(X,Y,X,R,S,T,TH,D), W = S, W1 = T$

/* no function, fit the basic safety convention of ATPL */

B. $ismem(X,Y,R,W,W1,TH,D) \leftarrow srr(Z,Y,X,R,S,T,TH,D), ismem(X,Z,R,W2,W3,TH,D1), dct(D,D1,W3,TH), W = W3*S, W1 = W3*T.$

/*the variable w3 in function item of rule header, is not limited extension of ATPL*/

C. $hasp(X,Y,P,W,W1,TH,D) \leftarrow ismem(X,Y,R,W,W1,TH,D), srp(X,P,R).$

/* no function, fit the basic safety convention of ATPL */

D. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(X,Y,X,P,S,T,TH,D), W = S, W1 = T.$

/* no function, fit the basic safety convention of ATPL */

E. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(Z,Y,X,P,S,T,TH,D), hasp(X,Z,P,W2,W3,TH,D1), dct(D,D1,W3,TH), W3*S, W1 = W3*T$

/* the variable w3 in function item of rule header, is not limited extension of ATPL */

F. $hasp(X,Y,P,W,W1,TH,D) \leftarrow sp(Z,Y,X,P,S,T,TH,D), ismem(X,Z,R,W2,W3,TH,D1), srp(X,P,R), dct(D,D1,W3,TH), W = W3*S, W1 = W3*T.$

/* all variable in function item of rule header, is limited extension of ATPL, which fits the basic safety convention so ATPL */

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Definition 7, the basic form of P_T: the structure of the $P_{T\to\Omega}$, which is the basic form of logic program P_T with trust built in

1) Replace the built-trust certificate, delegation authorization rules and authorization rules with alternative basic rules of basic form.

2) Eliminate other predicate related to trust value from the basic rules and conditional rules of step 1)

3) If the rule header of step 1) is empty, eliminate it.

According to the definition of Φ_T, its trunk predicate includes ismem, hasp, sp, srp and dct. Beside trunk predicate, ATPL can add condition predicates to delegation authorization based on demand. Here we assume the delegation authorization rule set with condition predicates added is Φ'_T. Because the ATPL is based on limited logic, and delegation authorization added obeys ATPL safety convention, it makes the antecedent of derivation morestrict, thus reduces the possible logic deduction that the rule sets arise. If Φ_T which only contains trunk predicate is safe, then $\Phi'_T$is also safe.

Elimination of condition predicates of P_T firstly, we eliminate the conditional rule in P_T and $P_{T\to\Omega}$. Secondly we eliminate the condition predicates in certificate and other authorization rules. Then we eliminate other condition predicates which have nothing to do with trust value in delegation authorization rules. Finally, we will get the corresponding reduction rules set $P'_T$and $P'_{T\to\Omega}$.

Theorem 4.10the basic form of logic program P'_T with built-in trust is $P'_{T\to\Omega}$,and the corresponding form of condition predicates elimination is P_T and $P_{T\to\Omega}$. Assuming the minimum Herbrand model of $P_{T\to\Omega}$ is M_Ω, then P_T has minimum model M_T, and $M_T \leq |M_\Omega|$.

Proof:According to the definition of ATPL rules, P_T and $P_{T\to\Omega}$ are consists of two parts of rules:

4) Certificate: rule header is predicate srr, sp

5) Delegation rules:

a) delegation authorization rule: rule header is predicate ismem, hasp

b) other authorization rule: rule header is predicate srp, srr, sp

So M_Ω contains the closed formulas constructed by the rule head predicate above. Assuming the logical inference set of P'_T is M_T', which is calculated by the inference operator Tp_T. Then M_T' is also consist of closed formulas constructed by the rule head predicate above.

Premise discussion:

According to the safety discussion of ATPL rules, the rule header predicate dct(D,D1,T,TH) in trust delegation depth control rules, appears in the built-in trust delegation rule set B, E, F. B and E are the only recursive rules with function in rule header in ATPL. So, according to the definition of delegation authorization, the inference with rule header are ismem and hasp, may leads to unlimited logical inference collection.

The definition of predicate of ismem and hasp are:

ismem(x:I_S, y:I_S, r:I_{SR}, w:IF_T, w1:I_T, th:I_T, d: I_D)

hasp(x:I_S, y:I_S, p:I_P, w:IF_T, w1:I_T, th:I_T, d: I_D)

To an authoritative source subject, the definable object set can is limited, the operation set on object is also limited. So on the premise of all rules fit the safety convention of ATPL, the rule with srp as rule header won't lead to infinite derivation. To srr and sp, we have two assumptions: (1) this discussion is based on a limited credentials set. (2) rule in local rule set with srr and sp as fit comply with the foregoing security convention.

Proposition:

From the construction process, we know that one-to-one relationship exists in the rule of simplified P_T and $P_{T\to\Omega}$. Here we make ordinal power operation of inferences operator to simplified P_T and $P_{T\to\Omega}$ (mark the delegation authorization in Φ as A'~F', related to the delegation authorization A~F in Φ_T):

1) $T_{P_T}\uparrow 0 = T_{P_{T\to\Omega}}\uparrow 0 = \phi$

2) Assume $T_{P_{T\to\Omega}}\uparrow 1 = \Theta_0, T_{P_T}\uparrow 1 = \Theta_0'$,To $\forall p \in \Theta_0$ exist $"p \leftarrow " \in P_{T\to\Omega}$,from the rule definition, we know {ismem,hasp}$\notin \Theta_0$,and{ismem,hasp}$\notin \Theta_0'$,then by the construction process of $P_{T\to\Omega}$ and P_T,we know that $\exists$ unique rule $"p' \leftarrow " \in P_T$ correspondences with $"p \leftarrow " \in P_{T\to\Omega}$. p' is the built-in trust extension form of p then $p' \in \Theta_0'$,otherwise to $\forall q' \in \Theta_0', \exists$ unique closed formula $q \in \Theta_0$,we have $|\Theta_0| = |\Theta_0'|$.

3) Assume $T_{P_{T\to\Omega}}\uparrow 2 = T_{P_{T\to\Omega}}\left(T_{P_{T\to\Omega}}\uparrow 1\right) = \Theta_1, T_{P_T}\uparrow 2 = T_{P_T}\left(T_{P_T}\uparrow 1\right) = \Theta_1'$,from 2) and the rule set Φ_T's definition, we know the deduction of has nothing to do with the rule B, E, F in Φ_T, so it's safe, Similar to 2), we can get Θ_1' ,and $|\Theta_1| = |\Theta_1'|$.

4) Assume $T_{P_{T\to\Omega}}\uparrow 3 = T_{P_{T\to\Omega}}\left(T_{P_{T\to\Omega}}\uparrow 2\right) =$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$\Theta_2, T_{P_T} \uparrow 3 = T_{P_T}(T_{P_T} \uparrow 2) = \Theta_2', |\Theta_2'| > |\Theta_2|$ will appear in the following case:

c) $\exists Q' \in \Theta_2'$,but $Q \notin \Theta_2$,According to the rule definition and safety analyze, we know $Q' \in \{ismem, hasp\}$, here we assume Q'=ismem(a,b,r,w,w1,th,d),then

(1) on the basis of delegation authorization rule A in Φ_T ,we get $\exists$ closed formula $srr(a,b,a,r,w,w1,th,d) \in \Theta_1'$,according to 3), we get: $\exists srr(a,b,a,r) \in \Theta_1$; On the basis of delegation authorization rule A',we get $ismem(a,b,r) \in \Theta_2$. So the precondition is wrong.

(2) From the delegation authorization rule B in Φ_T ,we get $\exists$ closed formula $(srr(g,b,a,r,t0,t1,th,d), ismem(a,g,r,w2,w3,th,d1)) \in \Theta_1'$ and the value of d,d1,w3,th fit delegation depth control rules; based on 3), we get: $\exists(srr(g,b,a,r), ismem(a,g,r)) \in \Theta_1$, From delegation authorization rule B',we get $ismem(a,b,r) \in \Theta_2$. So the precondition is wrong.

d) $\exists Q \in \Theta_2$,exist n>1 built-in trust $Q' \in \Theta_2'$ .From the rule definition, we know $Q \in \{ismem, hasp\}$,here we assume Q=ismem(a,b,r),then

(1) According to the delegation authorization rule A'in Φ , $\exists$ closed formula $srr(a,b,a,r) \in \Theta_1$.

From 3), we get: $\exists srr(a,b,a,r,t0,t1,th,d) \in \Theta_2'$. From the rule A' in Φ_T ,we can get unique Q'=ismem(a,b,r,t0,t1,th,d) $\in \Theta_2'$,So the precondition is wrong.

(2) According to the delegation authorization rule B'in Φ , $\exists$ closed formula $(srr(g,b,a,r), ismem(a,g,r)) \in \Theta_1$,from 3) we know $\exists(srr(g,b,a,r,t0,t1,th,d), ismem(a,g,r,w2,w3,th,d1)) \in \Theta_1'$, respectively. When the value of d,d1,w3,th does not fit delegation depth control predicates,no corresponding Q' generates; otherwise, we can calculate $f_m(w3,t0)$ and $f_m(w3,t1)$, and get unique w and w1,then we can get unique $ismem(a,b,r,w,w1,th,d) \in \Theta_2'$,So the precondition is wrong.

We can prove hasp in the same way. In a word,we have $|\Theta_2'| \leq |\Theta_2|$.

5) Assume $T_{P_T \to \Omega} \uparrow i = T_{P_T \to \Omega}(T_{P_T \to \Omega} \uparrow (i-1)) = \Theta_{i-1}, T_{P_T} \uparrow i = T_{P_T}(T_{P_T} \uparrow (i-1)) = \Theta_{i-1}$,then $|\Theta_{i-1}'| \leq |\Theta_{i-1}|$,to $T_{P_T \to \Omega} \uparrow (i+1) = T_{P_T \to \Omega}(T_{P_T \to \Omega} \uparrow i) = \Theta_i$ and $T_{P_T} \uparrow (i+1) = T_{P_T}(T_{P_T} \uparrow i)) = \Theta_i'$, in the same way of 4), we can prove $|\Theta_i'| \leq |\Theta_i|$

6) Because $P_{T \to \Omega}$ fit the ATPL safety convention ATPL, so least fixed point exists in its logical inference set.

$T_{P_T \to \Omega} \uparrow \omega = T_{P_T \to \Omega} \uparrow n = T_{P_T \to \Omega}(T_{P_T \to \Omega} \uparrow (n-1)) = M\Omega$

From the deduction 1)-5), we know that in the rule B,E,F of Φ_T ,when the delegation depth control predicate is matched,its variable parameters have been radical, so there is no variable left, and it's safe. The inference computing symbol $T_P(I)$ is monotonic, so $\forall i > j > 0$ and $|\Theta_j'| \leq |\Theta_i'| \leq |T_{P_T} \uparrow \omega|$. When P_T contains one or more rules in B, E, F, with the effect of delegation depth control predicates,we have $|\Theta_i'| \leq |\Theta_i|$ (0<=i),then $|T_{P_T} \uparrow \omega| \leq |T_{P_T \to \Omega} \uparrow \omega| = M\Omega$ so $|\Theta_0'| \leq |\Theta_1'| \leq \ldots \leq |T_{P_T} \uparrow \omega| \leq M\Omega$,ATPL is based on limited logic, and has no negative literal, So P_T's minimal model M_T is $T_{P_T} \uparrow \omega$ and $|M_T| \leq |M\Omega|$.

The proof is over.

6. CONCLUSION

Based on TEAMA and Constraint Datalog, this article defines a Policy Language for Trust management in Ad hoc networks, ATPL, realizes the transfer and calculation of the trust value during the trust delegation. This article also defines the syntax of ATPL, and discusses the security problem of ATPL. Firstly, we propose basic safety convention of ATPL rules based on basic safety convention of Datalog. Then we propose the extended safety convention of ATPL, and restrict the constraint domain into several constraint domain, by analyzing the part of rules in ATPL which does not match the basic safety convention. Finally, we discuss the semantics of ATPL, based on explanatory semantics of Datalog, work out the safety problem left in the delegation depth rule set and delegation authorization rule set, and prove that ATPL logic program is safe.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ACKNOWLEDGEMENTS

LI CHENG is the corresponding author of this paper.

REFERENCES:

[1] Fei Hu, Neeraj K. Sharma, "Security considerations in ad hoc sensor networks", *Ad Hoc Networks*, Vol. 3, No. 1, 2005, pp. 69-89.

[2] Kim Langfield-Smith, David Smith, "Management control systems and trust in outsourcing relationships", *Management Accounting Research*, Vol.14, No. 3, 2003, pp. 281-307.

[3] Qinghai Bai, "Study on the access control model", *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)*, IEEE Conference Publishing Services, July 26-30, 2010, pp. 830-834.

[4] Andrea Calì, Georg Gottlob, Thomas Lukasiewicz, "A general Datalog-based framework for tractable query answering over ontologies", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 14, 2012, pp. 57-83.

[5] Jin-Hee Cho, Ananthram Swami, Ing-Ray Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks", *Journal of Network and Computer Applications*, Vol. 35, No. 3, 2012, pp. 1001-1012.

[6] Sini Ruohomaa, Lea Kutvonen, "Trust Management Survey", Third International Conference, Springer Berlin Heidelberg, May 23-26, 2005, pp. 77-92.

[7] Basit Qureshi, Geyong Min, Demetres Kouvatsos, "A distributed reputation and trust management scheme for mobile peer-to-peer networks", *Computer Communications*, Vol. 35, No. 5, 2012, pp. 608–618.

[8] Zhengde Zhai, Dengguo Feng, Zhen Xu, "Fine-Grained Controllable Delegation Authorization Model Based on Trustworthiness", *Journal of Software*, Vol.18, No.8, 2007, pp. 2002-2015.

[9] Qi Jing, Jianbin Hu, Zhi Guan, Zhong Chen, "TEAMA: Trust Evaluation Based Authorization Model for Ad Hoc Networks", *Proceedings of the 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, IEEE Computer Society, October 21-23, 2011, pp. 470-474..

[10] Blaze M, Feigenbaum J, Lacy J., "Decentralized Trust Management", *Proc. of 17th Symposium on Security and Privacy*, IEEE Computer Society Press, May 1996, pp. 164-173.

[11] Clarke D, Elien J, Ellison C, Fredette M, Morcos A, Rivest R, "Certificate Chain Discovery in SPKI/SDSI", *Journal of Computer Security*, Vol. 9, No. 4, 2001, pp. 285-322.

[12] Elisa Bertino, Piero Andrea Bonatti,Elena Ferrari, "TRBAC: A Temporal Role-Based Access Control Model", *ACM Transactions on Information and System Security*, Vol. 4, No. 3, 2001, pp.191–223.

[13] Jie Jiang, Junhui Xiang, Hua Zhou, Xiaolin Zheng, Tianyang Dong, "Trust Calculation Model Based on Social Network and Evidence Theory", IEEE Conference Publishing Services, May 25-27, 2011, pp. 173-177