e-Informatica Software Engineering Journal, Volume 4, Issue 1, 2010

Deriving RT^T Credentials for Role-Based Trust Management

Anna Felkner*, Krzysztof Sacha**
*Research and Academic Computer Network
** Warsaw University of Technology
anna.felkner@gmail.com, k.sacha@ia.pw.edu.pl

Abstract

Role-based trust management languages define a formalism, which uses credentials to handle trust in decentralized, distributed access control systems. A credential provides information about the privileges of users and the security policies issued by one or more trusted authorities. The main topic of this paper is RT^T, a language which supports manifold roles and role-product operators to express threshold and separation of duties policies. The core part of the paper defines a relational, set-theoretic semantics for the language, and introduces a deductive system, in which credentials can be derived from an initial set of credentials using a set of inference rules. The soundness and the completeness of the deductive system with respect to the semantics of RT^T is proved.

1. Introduction

The problem of guaranteeing that confidential data and services offered by a computer system are not made available to unauthorized users is a challenging issue, which must be solved by reliable software technologies that are used for building high-integrity applications. The traditional solution to this problem is an implementation of some access control techniques, by which users are identified, and granted or denied access to a system data and other resources, depending on their individual or group identity. The examples of such solutions can be Mandatory Access Control (MAC) facilities, Discretionary Access Control (DAC) and Role-Based Access Control (RBAC) systems. Such an approach fits well into closed and centralized environments, in which the identity of users is known in advance.

Quite new challenges arise in decentralized and open systems, where the identity of users is not known in advance and the set of users can change. For example, consider a university, in which the students are enrolled and registered in particular faculties, and no central registry of all the students of that university exists. The policy of the university is such that a student is eligible to attend a lecture given by a faculty, regardless of the faculty in which he or she is actually registered. However, how could a faculty (the lecture owner) know that Peter Pan is eligible to attend the lecture, if his name is unknown to this faculty? The identity of the student itself does not help in making a decision whether he or she is eligible to attend or not. What is needed to make such a decision is information about the privileges assigned to Peter Pan by other authorities (is he registered in a faculty), as well as trust information about the authority itself (is the faculty a part of this university).

Trust-management system is a standardized solution for controlling security-critical services in high-integrity applications (Figure 1). It helps answer questions related to the conformance of potentially dangerous operations to a security policy of an organization, and provides the users with a language for writing the policies and controlling access to system services and

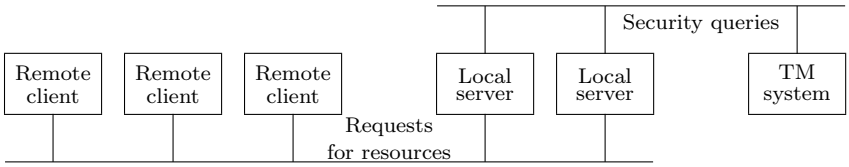

Figure 1. Trust management system

resources. The policies are no longer hard-coded into applications and therefore can be much easier to change. A designer of an application must only identify the security issues in the application and formulate appropriate queries to the trust-management system.

Such a conception of trust management, introduced in [2], has evolved since that time to a much broader context of assessing the reliability and developing trustworthiness for other systems and individuals [9]. In this paper, however, we will use the term trust management only in a meaning restricted to the field of access control.

The paper is organized as follows. An overview of the work related to role-based trust management systems and languages is given in Section 2. Section 4 describes the relational semantics of RT^T language. Section 6, which is the core part of our contribution, presents a deductive system, in which credentials can be derived from an initial set of credentials using a set of inference rules. A proof of the soundness and the completeness of the deductive system with respect to the semantics of RT^T is presented as well. Sections 3 and 5 provide the reader with illustrative examples. Final remarks and plans for future research are given in conclusions.

2. Related Work

Traditional access control systems usually rely on Role-Based Access Control model [14, 6, 7], which groups the access rights by the role name and limits the access to a resource to those users, who are assigned to a particular role. RBAC systems provide authorization decisions based on the identity of the users, and work well in centralized environment of an enterprise.

Trust management model represents quite another approach to access control, in which decisions are based on credentials (certificates) issued by multiple principals. A credential is an attestation of qualification, competence or authority, issued to an individual by a third party. Examples of credentials in real life include identification documents, social security cards, driver's licenses, membership cards, academic diplomas, certifications, security clearances, passwords and user names, keys, etc. A credential in a computer system can be a digitally signed document.

The potential and flexibility of trust management approach stems from the possibility of delegation: A principal may transfer limited authority over a resource to other principals. Such a delegation can be implemented by means of an appropriate credential. This way, a set of credentials can define the access control strategy and allow of deciding on who is authorized to access a resource, and who is not. A side-effect of delegation is such that a number of authorizing principals can be distributed over a network. A variety of problems arises if the credentials are stored in a decentralized manner.

The term trust management was first applied in the context of distributed access control in [2]. The first trust management system described in the literature was PolicyMaker [3], which defined a special assertion language capable of expressing policy statements, which were locally trusted, and credentials, which had to be signed using a private key. The next generation of trust management languages were KeyNote [1], which was an enhanced version of PolicyMaker, SPKI/SDSI [4] and a few other languages. All those languages allowed assigning privileges to entities and used credentials to delegate permissions from its issuer to its subject. What was missing in those languages was the possibility of delegation based on attributes of the entities and not on their identity.

Role-Based Trust management (RT) languages use roles to represent attributes [12]. The meaning of a role is a set of entities who have the attribute represented by the role. This meaning of roles captures the notion of groups of users in many systems and has been borrowed from Role-Based Access Control approach. The core language of RT family is RT_0, described in detail in [13]. It allows describing localized authorities for roles, role hierarchies, delegation of authority over roles and role intersections. All the subsequent languages add new features to RT_0.

RT_1 introduces parametrized roles, i.e. roles that are described using additional parameters, which can represent relationships between entities. RT_2 adds to RT_1 logical objects, which can represent permissions given to entities with respect to groups of logically related objects (resources). Those extensions can help in keeping the notation concise, but does not increase the expressive power of the language, because each combination of parameters in RT_1 and each permission to a logical object in RT_2 can be defined alternatively as a separate role in RT_0.

RT^T provides manifold roles and role-product operators, which can express threshold and separation of duties policies. A manifold role is a role that can be satisfied by a set of cooperating entities. A singleton role can be treated as a special case of a manifold role, whose set of cooperating entities is a singleton set. This way, RT_0 credentials can also be expressed in RT^T. A threshold policy requires a specified minimum number of entities to agree on some fact, e.g. in a requirement that two different bank cashiers must authorize a transaction. Separation of duties policy requires a set of entities, each of which fulfils a specific role, to agree before access is granted. Both types of policies mean that some transactions cannot be completed by a single entity, because no single entity has all the access rights required to complete the transaction.

RT^D provides mechanisms to describe delegation of role activations and selective use of role membership. This language is not covered in this paper. The features of RT^T and RT^D can be combined together with the features of RT_0, RT_1 or RT_2. A more detailed treatment of the Role-Based Trust management family of languages can be found in [12].

2.1. The Language RT_0

Basic elements of all the RT languages are entities, role names, roles and credentials. **Entities** represent principals that can define roles and issue credentials, and requesters that can make requests to access resources. An entity can be identified by a user account in a computer system or by a public key. **Role names** represent permissions that can be issued by entities to other entities or groups of entities. **Roles** represent sets of entities that have permissions issued by particular issuers. A role is defined as a pair composed of an entity (role issuer) and a role name. **Credentials** define roles by pointing a new member of the role or by delegating authority to the members of other roles.

In this paper, we use nouns beginning with a capital letter or just capital letters, e.g. A, B, C, to denote entities and sets of entities. Role names are denoted as identifiers beginning with a small letter or just small letters, e.g. r, s, t. Roles take the form of an entity (the issuer of this role) followed by a role name separated by a dot, e.g. $A.r$. Credentials are statements in the language. A credential consists of a role, left arrow symbol and a role expression.

There are four types of credentials in RT_0, which should be interpreted in the following way:

- $A.r \leftarrow B$ – *simple membership*: Entity B is a member of role $A.r$.
- $A.r \leftarrow B.s$ – *simple inclusion*: Role $A.r$ includes (all members of) role $B.s$. This is a delegation of authority over r from A to B, because B may cause new entities to become members of the role $A.r$ by issuing credentials that define $B.s$.
- $A.r \leftarrow B.s.t$ – *linking inclusion*: Role $A.r$ includes role $C.t$ for each C, which is a member of role $B.s$. This is a delegation of authority from A to all the members of the role $B.s$. The expression $B.s.t$ is called a *linked role*.
- $A.r \leftarrow B.s \cap C.t$ – *intersection inclusion*: Role $A.r$ includes all the entities who are members

of both roles $B.s$ and $C.t$. This is a partial delegation from A to B and C. The expression $B.s \cap C.t$ is called an *intersection role.*

A formal, set-theoretic semantics of RT_0 has been defined in a slightly different manner in [13] and [8].

Let $\mathcal{E}$ be a set of entities, $\mathcal{R}$ a set of role names and $\mathcal{P}$ a set of RT_0 credentials. The semantics of the set $\mathcal{P}$ of RT_0 credentials is a function $\mathcal{S}_\mathcal{P}$:

$$\mathcal{S}_\mathcal{P} : \mathcal{E} \times \mathcal{R} \rightarrow 2^\mathcal{E}.$$

such that $\mathcal{S}_\mathcal{P}$ is the least fixpoint of the following sequence of functions R_i, which map roles to sets of entity names [8]:

1. R_0 maps each role to an empty set ϕ
2. $R_{i+1} = \bigoplus_{c \in \mathcal{P}} f(R_i, c)$

where $\bigoplus$ is the point-wise extension of a function and f is a function that, given a (partial) semantics R_i and a credential $A.r \leftarrow e$, returns all the entities that should be added to $R_i(A.r)$, as governed by e:

$$f(R_i, A.r \leftarrow B) = \{A.r \mapsto \{B\}\}$$
$$f(R_i, A.r \leftarrow B.s) = \{A.r \mapsto R_i(B.s)\}$$
$$f(R_i, A.r \leftarrow B.s.t) = \{A.r \mapsto \bigcup_{C \in R_i(B.s)} R_i(C.t)\}$$
$$f(R_i, A.r \leftarrow B.s \cap C.t) = \{A.r \mapsto R_i(B.s) \cap R_i(C.t)\}$$

2.2. The Language RT^T

At the syntax level, RT^T adopts all the four types of RT_0 credentials, and adds two new types of credentials. These are:

$A.r \leftarrow B.s \odot C.t$ – role $A.r$ includes one member of role $B.s$ and one member of role $C.t$. This allows expressing threshold policies.

$A.r \leftarrow B.s \otimes C.t$ – role $A.r$ includes one member of role $B.s$ and one member of role $C.t$, but those members of roles have to be different. This allows for expressing separation of duties policies.

The changes at the semantics level are greater, because the requesters as well as the issuers of RT^T credentials are no longer entities, but sets of entities, who can jointly fulfil a role. Such a change applies to all six types of credentials, also those, which are adopted from RT_0.

Formal definition of the semantics of RT^T is covered in Section 4.

3. Examples

The models discussed in this paper can be, in general, very complex. Therefore, we present here only simplified examples, with the intention to illustrate the basic notions and the notation. The first example demonstrates the use of RT_0 credentials, while the second one presents the use of RT^T credentials.

Example 1 (RT_0)

A person has the right to attend a *lecture*, given at a university U, when he or she is a *student* registered to a faculty of this university. To be able to fulfil the role of a $faculty$, an organization ought to be a *division* of the university and should conduct *research* activities. *John* is a student registered to F, which is a *division* of U, and which conducts *research* activities. The following credentials prove that *John* have the right to attend a *lecture*:

$$U.lecture \leftarrow U.faculty.student \quad (1)$$
$$U.faculty \leftarrow U.division \cap U.research \quad (2)$$
$$U.division \leftarrow F \quad (3)$$
$$U.research \leftarrow F \quad (4)$$
$$F.student \leftarrow John \quad (5)$$

Example 2 (RT^T)

The following example has been adopted from [11]. A bank B has three roles: *manager*, *cashier* and *auditor*. Security policy of the bank requires an *approval* of certain transactions from a *manager*, two *cashiers*, and an *auditor*. The two *cashiers* must be different. However, a *manager* who is also a *cashier* can serve as one of the two cashiers. The *auditor* must be different from the other parties in the transaction.

Such a policy can be described using the following credentials:

$$B.twoCashiers \leftarrow B.cashier \otimes B.cashier \quad (6)$$

$$B.managerCashiers \leftarrow B.manager \odot B.twoCashiers \quad (7)$$

$$B.approval \leftarrow B.auditor \otimes B.managerCashiers \quad (8)$$

Now, assume that the following credentials have been added:

$$B.cashier \leftarrow Mary \quad (9)$$

$$B.cashier \leftarrow Doris \quad (10)$$

$$B.cashier \leftarrow Alice \quad (11)$$

$$B.cashier \leftarrow Kate \quad (12)$$

$$B.manager \leftarrow Alice \quad (13)$$

$$B.auditor \leftarrow Kate \quad (14)$$

Then one can conclude that, according to the policy of B, the following sets of entities can cooperatively approve a transaction: $\{Mary, Doris, Alice, Kate\}$, $\{Mary, Alice, Kate\}$ and $\{Doris, Alice, Kate\}$.

4. The Semantics of RT^T

The syntax of a language defines language expressions, which are used to communicate information. The primary expressions of Role-Based Trust management languages are credentials and sets of credentials, which are used as a means for defining roles.

The semantics of a language defines the meaning of expressions. Such a definition consists of two parts [10]: A semantic domain and a semantic mapping from the syntax to the semantic domain. The meaning of a language expression must be an element in the semantic domain.

The semantics of RT_0, which defines the meaning of a set of credentials as a function from the set of roles into the power set of entities, has no potential to describe the meaning of RT^T, which supports manifold roles and role-product operators. Therefore, we define in this section the meaning of a set of credentials as a relation over the set of roles and the power set of entities. Thus, we use a Cartesian product of the set of roles and the power set of entities as the semantic domain of a Role-Based Trust management language. The semantic mapping would associate a specific relation between roles and entities with each set of credentials. Such a relational approach allows us to define a formal semantics of RT^T language [5].

Let $\mathcal{E}$ be the set of entities and $\mathcal{R}$ be the set of role names. $\mathcal{P}$ is a set of RT-credentials, which describe the assignment of sets of entities to roles, issued by other entities (or rather sets of entities).

The semantics of $\mathcal{P}$, denoted by $\mathcal{S}_\mathcal{P}$, is defined as a relation:

$$\mathcal{S}_\mathcal{P} \subseteq 2^\mathcal{E} \times \mathcal{R} \times 2^\mathcal{E},$$

An instance of this relation, e.g.: (A, r, X), maps the role $A.r$ to a set of entities $X \in 2^\mathcal{E}$. If the cardinality of set X is greater than one, then the role $A.r$ is a manifold role and the entities of set X must cooperate together in order to satisfy the role. The cardinality of set A can also be greater than one, which would mean that the role $A.r$ is governed jointly by the entities of set A.

If all the sets of entities are singleton sets, the semantics of RT^T reduces to the semantics of RT_0. This way, our definition covers all the RT languages including RT_0 through RT^T.

Denote the power set of entities by $\mathcal{F} = 2^\mathcal{E}$. Each element in $\mathcal{F}$ is a set of entities from $\mathcal{E}$ (a subset of $\mathcal{E}$). Each element in $2^\mathcal{F}$ is a set, compound of sets of entities from $\mathcal{E}$.

The semantics of $\mathcal{P}$ can now be described in an alternative way as a function:

$$\widetilde{\mathcal{S}}_\mathcal{P} : 2^\mathcal{E} \times \mathcal{R} \rightarrow 2^\mathcal{F}$$

which maps each role from $2^\mathcal{E} \times \mathcal{R}$ into a set of subsets of entities. The members of each subset must cooperate in order to satisfy the role.

Knowing the relation $\mathcal{S}_\mathcal{P}$, one can define the function $\widetilde{\mathcal{S}}_\mathcal{P}$ as follows:

$$\widetilde{\mathcal{S}}_\mathcal{P}(A, r) = \{X \in 2^\mathcal{E} : (A, r, X) \in \mathcal{S}_\mathcal{P}\}$$

The semantics of RT^T can now be defined formally in the following way.

Definition 1. *The semantics of a set $\mathcal{P}$ of RT^T credentials, denoted by $\mathcal{S}_\mathcal{P}$, is the smallest relation S_i, such that:*
1. $\mathcal{S}_0 = \phi$
2. $\mathcal{S}_{i+1} = \bigcup_{c\in\mathcal{P}} f(\mathcal{S}_i, c)$ for $i = 0, 1, \ldots$

which is closed with respect to function f, which describes the meaning of credentials in the following way (A, B, C, X, Y are sets of entities, may be singletons):

$$f(\mathcal{S}_i, A.r \leftarrow X) = \{(A, r, X)\} \quad (D_1)$$
$$f(\mathcal{S}_i, A.r \leftarrow B.s) = \{(A, r, X) : (B, s, X) \in \mathcal{S}_i\} \quad (D_2)$$
$$f(\mathcal{S}_i, A.r \leftarrow B.s.t) = \bigcup_{C:(B,s,C)\in\mathcal{S}_i}\{(A, r, X) : (C, t, X) \in \mathcal{S}_i\} \quad (D_3)$$
$$f(\mathcal{S}_i, A.r \leftarrow B.s \cap C.t) = \{(A, r, X) : (B, s, X) \in \mathcal{S}_i \wedge (C, t, X) \in \mathcal{S}_i\} \quad (D_4)$$
$$f(\mathcal{S}_i, A.r \leftarrow B.s \odot C.t) = \{(A, r, X \cup Y) : (B, s, X) \in \mathcal{S}_i \wedge (C, t, Y) \in \mathcal{S}_i\} \quad (D_5)$$
$$f(\mathcal{S}_i, A.r \leftarrow B.s \otimes C.t) = \{(A, r, X \cup Y) : (B, s, X) \in \mathcal{S}_i \wedge (C, t, Y) \in \mathcal{S}_i \wedge (X \cap Y) = \phi\} \quad (D_6)$$

5. Examples

We use the example sets of credentials from Section 3 to illustrate the definition of RT^T semantics.

Example 1 (RT_0)

The starting relation $\mathcal{S}_0$ is, by definition, empty. The sequence of steps to compute consecutive relations $\mathcal{S}_i$ can be described as follows:
$\mathcal{S}_0 = \phi$
$\mathcal{S}_1 = \{(\{U\}, division, \{F\}),\ (\{U\}, research, \{F\}),\ (\{F\}, student, \{John\})\}$
$\mathcal{S}_2 = \{(\{U\}, division, \{F\}),\ (\{U\}, research, \{F\}),\ (\{F\}, student, \{John\}),$
$(\{U\}, faculty, \{F\})\}$
$\mathcal{S}_3 = \{(\{U\}, division, \{F\}),\ (\{U\}, research, \{F\}),\ (\{F\}, student, \{John\}),$
$(\{U\}, faculty, \{F\}),\ (\{U\}, lecture, \{John\})\}$

The resulting relation $\mathcal{S}_3$ cannot be changed using the given set of credentials, hence: $\mathcal{S}_P = \mathcal{S}_3$. Because the RT language considered in this example is RT_0, all the sets of entities are singleton sets.

Example 2 (RT^T)

The sequence of steps to compute consecutive relations $\mathcal{S}_i$ starts from an empty set, $\mathcal{S}_0 = \phi$, and proceeds as follows. Credentials 9 through 14 are mapped in $\mathcal{S}_0$ into relation $\mathcal{S}_1$:
$\mathcal{S}_1 = \{(\{B\}, cashier, \{Mary\}),\ (\{B\}, cashier, \{Doris\}),$
$(\{B\}, cashier, \{Alice\}),\ (\{B\}, cashier, \{Kate\}),$
$(\{B\}, manager, \{Alice\}),\ (\{B\}, auditor, \{Kate\})\}$
Credential 6 adds the following instances to relation $\mathcal{S}_2$:
$\mathcal{S}_2 = \mathcal{S}_1 \cup \{$
$(\{B\}, twoCashiers, \{Mary, Doris\}),\ (\{B\}, twoCashiers, \{Mary, Alice\}),$
$(\{B\}, twoCashiers, \{Mary, Kate\}),\ (\{B\}, twoCashiers, \{Doris, Alice\}),$
$(\{B\}, twoCashiers, \{Doris, Kate\}),\ (\{B\}, twoCashiers, \{Alice, Kate\})\}$
Credentials 7 is resolved in $\mathcal{S}_3$:
$\mathcal{S}_3 = \mathcal{S}_2 \cup \{$
$(\{B\}, managerCashiers, \{Mary, Doris, Alice\}),$

$(\{B\}, managerCashiers, \{Mary, Alice\})$,
$(\{B\}, managerCashiers, \{Mary, Kate, Alice\})$,
$(\{B\}, managerCashiers, \{Doris, Alice\})$,
$(\{B\}, managerCashiers, \{Doris, Kate, Alice\})$,
$(\{B\}, managerCashiers, \{Alice, Kate\})\}$,
and credential 8 in $\mathcal{S}_4$:
$\mathcal{S}_4 = \mathcal{S}_3 \cup \{$
$(\{B\}, approval, \{Mary, Doris, Alice, Kate\})$,
$(\{B\}, approval, \{Mary, Alice, Kate\})$,
$(\{B\}, approval, \{Doris, Alice, Kate\})\}$,

The resulting relation $\mathcal{S}_4$ cannot be changed using the given set of credentials, hence: $\mathcal{S}_P = \mathcal{S}_4$. Because the RT language considered in this example is RT^T, there is a set of sets of entities assigned to each role.

6. Deductive system over RT^T credentials

RT^T credentials are used to define roles and roles are used to represent permissions. The semantics of a given set $\mathcal{P}$ of RT^T credentials defines for each role $A.r$ the set of entities which are members of this role. The member sets of roles can also be calculated in a more convenient way using a deductive system, which defines an operational semantics of RT^T language.

A **deductive system** consists of an initial set of formulae that are considered to be true, and a set of **inference rules**, that can be used to derive new formulae from the known ones.

Let $\mathcal{P}$ be a given set of RT^T credentials. The application of inference rules of the deductive system will create new credentials, derived from credentials of the set $\mathcal{P}$. A derived credential c will be denoted using a formula:

$$\mathcal{P} \succ c$$

which should be read: "credential c can be derived from a set of credentials $\mathcal{P}$".

Definition 2. *The initial set of formulae of a deductive system over a set $\mathcal{P}$ of RT^T credentials are all the formulae:*

$$c \in \mathcal{P}$$

for each credential c in $\mathcal{P}$. The inference rules of the system are the following:

$$\frac{c \in \mathcal{P}}{\mathcal{P} \succ c} \quad (W_1)$$

$$\frac{\mathcal{P} \succ A.r \leftarrow B.s \qquad \mathcal{P} \succ B.s \leftarrow X}{\mathcal{P} \succ A.r \leftarrow X} \quad (W_2)$$

$$\frac{\mathcal{P} \succ A.r \leftarrow B.s.t \qquad \mathcal{P} \succ B.s \leftarrow C \qquad \mathcal{P} \succ C.t \leftarrow X}{\mathcal{P} \succ A.r \leftarrow X} \quad (W_3)$$

$$\frac{\mathcal{P} \succ A.r \leftarrow B.s \cap C.t \qquad \mathcal{P} \succ B.s \leftarrow X \qquad \mathcal{P} \succ C.t \leftarrow X}{\mathcal{P} \succ A.r \leftarrow X} \quad (W_4)$$

$$\frac{\mathcal{P} \succ A.r \leftarrow B.s \odot C.t \qquad \mathcal{P} \succ B.s \leftarrow X \qquad \mathcal{P} \succ C.t \leftarrow Y}{\mathcal{P} \succ A.r \leftarrow X \cup Y} \quad (W_5)$$

$$\frac{\mathcal{P} \succ A.r \leftarrow B.s \otimes C.t \qquad \mathcal{P} \succ B.s \leftarrow X \qquad \mathcal{P} \succ C.t \leftarrow Y \qquad X \cap Y = \phi}{\mathcal{P} \succ A.r \leftarrow X \cup Y} \quad (W_6)$$

There could be a number of deductive systems defined over a given language. To be useful for practical purposes a deductive system must exhibit two properties. First, it should be sound, which means that the inference rules could derive only formulae that are valid with respect to the semantics of the language. Second, it should be complete, which means that each formula, which is valid according to the semantics, should be derivable in the system.

All the credentials, which can be derived in the system, either belong to set $\mathcal{P}$ (rule W_1) or are of the type: $\mathcal{P} \succ A.r \leftarrow X$ (rules W_2 through W_6). To prove the soundness of the deductive system, one must prove that for each new formula $\mathcal{P} \succ A.r \leftarrow X$, the triple (A, r, X) belongs to the semantics $\mathcal{S}_\mathcal{P}$ of the set $\mathcal{P}$.

Let we first note that all the formulae $\mathcal{P} \succ A.r \leftarrow X$, such that $A.r \leftarrow X \in P$ are sound. This is proved in Lemma 1.

Lemma 1. *If* $A.r \leftarrow X \in \mathcal{P}$ *then* $(A, r, X) \in \mathcal{S}_\mathcal{P}$.

Proof. The relation $\mathcal{S}_\mathcal{P}$, which defines the semantics of $\mathcal{P}$, is a limit of a monotonically increasing sequence of sets $S_0, S_1 \ldots$ such that $S_0 = \phi$. According to Definition 1: $f(S_0, A.r \leftarrow X) = (A, r, X)$ Hence, $(A, r, X) \in S_1$ and because $S_1 \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X) \in S_P$. □

To prove the soundness of the deductive system over $\mathcal{P}$, we must prove the soundness of each formula $\mathcal{P} \succ A.r \leftarrow X$, which can be derived from the set $\mathcal{P}$. This is proved in Theorem 1.

Theorem 1. *If* $\mathcal{P} \succ A.r \leftarrow X$ *then* $(A, r, X) \in \mathcal{S}_\mathcal{P}$.

Proof. By induction with respect to the number n of inference steps, which are needed to derive a formula $\mathcal{P} \succ A.r \leftarrow X$.

If $n = 1$ then the formula $\mathcal{P} \succ A.r \leftarrow X$ could be derived only using rule W_1, because the premises of only this rule belong to the initial set of formulae of the deductive system. Hence, the thesis is true according to Lemma 1.

Consider $n > 1$ and assume for the inductive step that the thesis is true if the number of inference steps was not greater than n. We will show that it is true also in a case when the number of inference steps equals $n + 1$.

Each of the rules W_2 through W_6 could be used in the last $(n + 1)$ step of inference. All those five cases are discussed separately.

[W_2] The first premise of W_2 cannot be derived otherwise than using W_1. Hence, $A.r \leftarrow B.s \in P$. The second premise of $W_2 : \mathcal{P} \succ B.s \leftarrow X$ was derived from $\mathcal{P}$ using at most n steps of inference, hence, $(B, s, X) \in \mathcal{S}_\mathcal{P}$ according to the inductive hypothesis. By Definition 1, there exists such $\mathcal{S}_i$ that $(B, s, X) \in \mathcal{S}_i$, and $(A, r, X) \in f(\mathcal{S}_i, A.r \leftarrow B.s)$ according to (D_2). Because $f(\mathcal{S}_i, A.r \leftarrow B.s) \subseteq \mathcal{S}_{i+1} \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X) \in \mathcal{S}_\mathcal{P}$.

[W_3] The first premise of W_3 cannot be derived otherwise than using W_1. Hence, $A.r \leftarrow B.s.t \in P$. The second premise of $W_3 : \mathcal{P} \succ B.s \leftarrow C$ was derived from $\mathcal{P}$ using at most n steps of inference, hence, $(B, s, C) \in \mathcal{S}_\mathcal{P}$ according to the inductive hypothesis. By Definition 1, there exists such $\mathcal{S}_i$ that $(B, s, C) \in \mathcal{S}_i$. Similarly, in the case of the third premise of $W_3 : \mathcal{P} \succ C.t \leftarrow X$, there exists such $\mathcal{S}_j$ that $(C, t, X) \in \mathcal{S}_j$. Let k be the maximum of (i, j). Then $(B, s, C) \in \mathcal{S}_k$ and $(C, t, X) \in \mathcal{S}_k$, and $(A, r, X) \in f(\mathcal{S}_k, A.r \leftarrow B.s.t)$ according to (D_3). Because $f(\mathcal{S}_k, A.r \leftarrow B.s.t) \subseteq \mathcal{S}_{k+1} \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X) \in \mathcal{S}_\mathcal{P}$.

[W_4] The first premise of W_4 cannot be derived otherwise than using W_1. Hence, $A.r \leftarrow B.s \cap C.t \in P$. The second premise of $W_4 : \mathcal{P} \succ B.s \leftarrow X$ was derived from $\mathcal{P}$ using at most n steps of inference, hence, $(B, s, X) \in \mathcal{S}_\mathcal{P}$ according to the inductive hypothesis. By Definition 1, there exists such $\mathcal{S}_i$ that $(B, s, X) \in \mathcal{S}_i$. Similarly, in the case of the third premise of $W_4 : \mathcal{P} \succ C.t \leftarrow X$, there exists such $\mathcal{S}_j$ that $(C, t, X) \in \mathcal{S}_j$. Let k be the maximum of (i, j). Then $(B, s, X) \in \mathcal{S}_k$, $(C, t, X) \in \mathcal{S}_k$ and $(A, r, X) \in f(\mathcal{S}_k, A.r \leftarrow B.s \cap C.t)$ according to (D_4). Because $f(\mathcal{S}_k, A.r \leftarrow B.s \cap C.t) \subseteq \mathcal{S}_{k+1} \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X) \in \mathcal{S}_\mathcal{P}$.

[W_5] The conclusion of W_5 is a formula $P \succ A.r \leftarrow X \odot Y$, which states that the set of entities that can play a role $A.r$ is a union of two another sets of entities X and Y. To prove the thesis we must show that $(A, r, X \cup Y) \in \mathcal{S}_\mathcal{P}$.

The first premise of W_5 cannot be derived otherwise than using W_1. Hence, $A.r \leftarrow B.s \odot$

$C.t \in \mathcal{P}$. Similarly as in case of W_4, the second and the third premises of W_5 were derived from $\mathcal{P}$ using at most n steps of inference. So, $(B, s, X) \in \mathcal{S}_\mathcal{P}$ and $(C, t, Y) \in \mathcal{S}_\mathcal{P}$. Then, there exists such k that $(B, s, X) \in \mathcal{S}_k$ and $(C, t, Y) \in \mathcal{S}_k$, and $(A, r, X \cup Y) \in f(\mathcal{S}_k, A.r \leftarrow B.s \odot C.t)$ according to (D_5). Because $f(\mathcal{S}_k, A.r \leftarrow B.s \odot C.t) \subseteq \mathcal{S}_{k+1} \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X \cup Y) \in \mathcal{S}_\mathcal{P}$.

[W_6] The conclusion of W_6 is a formula $P \succ A.r \leftarrow X \otimes Y$, which states that the set of entities that can play a role $A.r$ is a union of two another sets of entities X and Y. To prove the thesis we must show that $(A, r, X \cup Y) \in \mathcal{S}_\mathcal{P}$.

The first premise of W_6 cannot be derived otherwise than using W_1. Hence, $A.r \leftarrow B.s \otimes C.t \in \mathcal{P}$. Similarly as in case of W_4, the second and the third premises of W_6 were derived from $\mathcal{P}$ using at most n steps of inference. So, $(B, s, X) \in \mathcal{S}_\mathcal{P}$ and $(C, t, Y) \in \mathcal{S}_\mathcal{P}$. Then, there exists such k that $(B, s, X) \in \mathcal{S}_k$ and $(C, t, Y) \in \mathcal{S}_k$. The fourth premise of W_6: $X \cap Y = \phi$, does not depend on the number of inference steps and is always true if W_6 could be applied. Hence, $(A, r, X \cup Y) \in f(\mathcal{S}_k, A.r \leftarrow B.s \otimes C.t)$according to (D_6). Because $f(\mathcal{S}_k, A.r \leftarrow B.s \otimes C.t) \subseteq \mathcal{S}_{k+1} \subseteq \mathcal{S}_\mathcal{P}$ then $(A, r, X \cup Y) \in \mathcal{S}_\mathcal{P}$. □

To prove the completeness of the deductive system over a set $\mathcal{P}$ of RT^T credentials, we must prove that a formula $P \succ A.r \leftarrow X$ can be derived using inference rules for each element $(A, r, X) \in \mathcal{S}_\mathcal{P}$. This is proved in Theorem 2.

Theorem 2. *If $(A, r, X) \in \mathcal{S}_\mathcal{P}$ then $\mathcal{P} \succ A.r \leftarrow X$.*

Proof. Assume $(A, r, X) \in \mathcal{S}_\mathcal{P}$. By Definition 1, there exists such $i \geq 0$ and such $c \in \mathcal{P}$ that $(A, r, X) \in f(\mathcal{S}_i, c)$. The proof of the thesis is by induction with respect to the value of index i.

If $i = 0$ then credential c must take the form of $A.r \leftarrow X$. This is because $\mathcal{S}_0 = \phi$ and $f(\mathcal{S}_0, d) = \phi$ for each credential d other than $A.r \leftarrow X$. Hence, $A.r \leftarrow X \in \mathcal{P}$ and the formula $\mathcal{P} \succ A.r \leftarrow X$ can be derived using rule W_1.

Let $i > 0$. Assume for the inductive step that the thesis is true, if the value of index i in the expression $(A, s, X) \in f(\mathcal{S}_i, c)$ was not greater than n. We will show that it is true also in the case when the value of index i equals $n + 1$.

Assume $(A, r, X) \in \mathcal{S}_\mathcal{P}$ and $(A, r, X) \in f(\mathcal{S}_{n+1}, c)$ for a certain $c \in \mathcal{P}$. The credential c can take one of the six forms allowed in RT^T. Each of these types of credentials will be discussed separately.

[$c = A.r \leftarrow X$] If this is the case, then the formula $\mathcal{P} \succ A.r \leftarrow X$ can be derived using rule W_1.

[$c = A.r \leftarrow B.s$] If $(A, r, X) \in f(\mathcal{S}_{n+1}, A.r \leftarrow B.s)$, then $(B, s, X) \in \mathcal{S}_{n+1}$ according to (D_2) of Definition 1. Hence, there exists a credential $c \in \mathcal{P}$ such that $(B, s, X) \in f(\mathcal{S}_n, c)$. This implies that $(B, s, X) \in \mathcal{S}_\mathcal{P}$ and $\mathcal{P} \succ B.s \leftarrow X$ according to the inductive hypothesis. Then $\mathcal{P} \succ A.r \leftarrow B.s$ and $\mathcal{P} \succ B.s \leftarrow X$, hence, $\mathcal{P} \succ A.r \leftarrow X$ is a conclusion of rule W_2.

[$c = A.r \leftarrow B.s.t$] If $(A, r, X) \in f(\mathcal{S}_{n+1}, A.r \leftarrow B.s.t)$ then according to (D_3) of Definition 1, there exists a set of entities C such that $(B, s, C) \in \mathcal{S}_{n+1}$ and $(C, t, X) \in \mathcal{S}_{n+1}$. Hence, there exists a credential $c_1 \in \mathcal{P}$ such that $(B, s, C) \in f(\mathcal{S}_n, c_1)$ and there exists a credential $c_2 \in \mathcal{P}$ such that $(C, t, X) \in f(\mathcal{S}_n, c2)$. This implies that $(B, s, C) \in \mathcal{S}_\mathcal{P}$ and $(C, t, X) \in \mathcal{S}_\mathcal{P}$, hence, $\mathcal{P} \succ B.s \leftarrow C$ and $\mathcal{P} \succ C.t \leftarrow X$ according to the inductive hypothesis. $\mathcal{P} \succ A.r \leftarrow X$ is a conclusion of rule W_3.

[$c = A.r \leftarrow B.s \cap C.t$] If $(A, r, X) \in f(\mathcal{S}_{n+1}, A.r \leftarrow B.s \cap C.t)$ then $(B, s, X) \in \mathcal{S}_{n+1}$ and $(C, t, X) \in \mathcal{S}_{n+1}$ according to (D_4) of Definition 1. Hence, there exist credentials c_1, c_2 such that $(B, s, X) \in f(\mathcal{S}_n, c_1)$ and $(C, t, X) \in f(\mathcal{S}_n, c_2)$. This implies that $(B, s, X) \in \mathcal{S}_\mathcal{P}$ and $(C, t, X) \in \mathcal{S}_\mathcal{P}$, hence, $\mathcal{P} \succ B.s \leftarrow X$ and $\mathcal{P} \succ C.t \leftarrow X$ according to the inductive hypothesis. $\mathcal{P} \succ A.r \leftarrow X$ is a conclusion of rule W_4.

[$c = A.r \leftarrow B.s \odot C.t$] If $(A, r, X) \in f(\mathcal{S}_{n+1}, A.r \leftarrow B.s \odot C.t)$, then according to (D_5) of Definition 1, there exist two sets of entities Z, Y such that $Z \cup Y = X$ and $(B, s, Z) \in \mathcal{S}_{n+1}$ and $(C, t, Y) \in \mathcal{S}_{n+1}$. Hence, there exist credentials c_1, c_2 such that $(B, s, Z) \in f(\mathcal{S}_n, c_1)$ and $(C, t, Y) \in f(\mathcal{S}_n, c2)$. This implies that $(B, s, Z) \in \mathcal{S}_\mathcal{P}$ and $(C, t, Y) \in \mathcal{S}_\mathcal{P}$, hence, $\mathcal{P} \succ B.s \leftarrow Z$ and $\mathcal{P} \succ C.t \leftarrow Y$ according to the inductive hypothesis. $\mathcal{P} \succ A.r \leftarrow X$ is a conclusion of rule W_5.

$[\boldsymbol{c = A.r \leftarrow B.s \otimes C.t}]$ If $(A, s, X) \in f(\mathcal{S}_{n+1}, A.r \leftarrow B.s \otimes C.t)$, then according to (D_6) of Definition 1, there exist two sets of entities Z, Y such that $Z \cup Y = X$ and $Z \cap Y = \phi$ and $(B, s, Z) \in \mathcal{S}_{n+1}$ and $(C, t, X) \in \mathcal{S}_{n+1}$. Hence, there exist credentials c_1, c_2 such that $(B, s, Z) \in f(\mathcal{S}_n, c_1)$ and $(C, t, Y) \in f(\mathcal{S}_n, c_2)$. This implies that $(B, s, Z) \in \mathcal{S}_{\mathcal{P}}$ and $(C, t, Y) \in \mathcal{S}_{\mathcal{P}}$, hence, $\mathcal{P} \succ B.s \leftarrow Z$ and $\mathcal{P} \succ C.t \leftarrow Y$ according to the inductive hypothesis. $\mathcal{P} \succ A.r \leftarrow X$ is a conclusion of rule W_6. □

A conclusion from Theorem 1 and Theorem 2 is such that the deductive system of Definition 2 is sound and complete with respect to the semantics of RT^T credentials. This way, the deductive system gives an operational definition of RT^T semantics.

7. Conclusions

This paper deals with modelling of trust management systems in decentralized and distributed environments. The modelling framework is a family of Role-Based Trust management language RT^T. Two types of semantics for a set of RT^T credentials have been introduced in the paper.

A set-theoretic semantics of RT^T is defined as a relation over a set of roles and a power set (set of sets) of entities. All the members of a set of entities related to a role must cooperate in order to satisfy the role. This way, our definition covers the full potential of RT^T, which supports the notion of manifold roles and is able to express structure of threshold and separation-of-duty policies.

An operational semantics of RT^T is defined as a deductive system, in which credentials can be derived from an initial set of credentials using a set of inference rules. The semantics is given by the set of resulting credentials of the type $A.r \leftarrow X$, which explicitly show a mapping between roles and sets of entities.

The properties of soundness and completeness of the deductive system with respect to the semantics of RT^T are proved.

References

[1] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming*, pages 185–210. 1999.

[2] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the IEEE Conference on Security and Privacy*, pages 164–173, 1996.

[3] M. Blaze, J. Feigenbaum, and M. Strauss. Compliance checking in the PolicyMaker trust management system. In *Financial Cryptography*, pages 1439–1456, 1998.

[4] D. Clarke, J. E. Elienb, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.

[5] A. Felkner and K. Sacha. The semantics of role-based trust management languages. In *Proc. Central and Eastern European Conference on Software Engineering Techniques CEE-SET*, pages 195–206, 2009.

[6] D. Ferraiolo and D. Kuhn. Role-based access control. In *Proc. 15th National Computer Security Conference*, pages 554–563, 1992.

[7] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):224–274, 2001.

[8] D. Gorla, M. Hennessy, and V. Sassone. Inferring dynamic credentials for role-based trust management. In *Proceedings of the 8th ACM SIGPLAN international conference on Principles and practice of declarative programming*, page 224, 2006.

[9] W. M. Grudzewski, I. K. Hejduk, A. Sankowska, and M. Wantuchowicz. *Trust Management in Virtual Work Environments: A Human Factors Perspective*. CRC Press, 2008.

[10] D. Harel and B. Rumpe. Modeling languages: Syntax, semantics and all that stu. 2000.

[11] N. Li and J. Mitchell. RT: a role-based trust-management framework. In *Proc. 3rd DARPA Information Survivability Conference*

and Exposition, pages 201–212. IEEE Computer Society Press, 2003.

[12] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust-management framework. In *Proceedings of 2002 IEEE Symposium on Security and Privacy*, pages 114–130, Oakland CA, 2002. IEEE Computer Society Press.

[13] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, 2003.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.