The Equivalent Conversions of the Role-Based Access Control Model

Nadezda Bogachenko
Dostoevsky Omsk State University
Omsk, Russia
nfbogachenko@mail.ru

***Abstract* — The problems which are important for the effective functioning of an access control policy in a large information system (LIS) are selected. The general concept of a local optimization of a role-based access control (RBAC) model is formulated. The optimization criteria are proposed. The algorithms of a local optimization of the RBAC model are defined and justified. The developed algorithms are used in the methods of the solution of the following problems: the assessment of risks of the leakage of permissions in the RBAC policy, the access control in the distributed hierarchical systems, the combining of role-based and mandatory access control models. In the first problem the question of the permissions distribution in the role hierarchy is researched. The analytic hierarchy process (AHP) is applied to creation of the estimates. The method is based on the hierarchical structure of a role set. The offered technique can order the permissions according to the value of the risks of their leakage. In the second problem the algorithm of the distribution of the cryptographic keys in the system with a hierarchical arrangement of the objects is offered. The cryptography protocols for the practical use of this algorithm are defined. The conditions of the implementation of the discretionary and mandatory principles of the access control on the basis of the developed algorithm are formulated.**

Keywords — access control; role hierarchy; local optimization; permissions leakage risks; analytic hierarchy process; hash function; sequential access.

I. Introduction

The research objective is the development of methods and algorithms of the solution of the problems which arise in the case of the implementation of an access control policy in a LIS. The specifics of the data access arrangement in a LIS are defined. Here are the most significant factors:

- The demand of the RBAC mechanisms as the object-oriented decision which is capable to reduce complexity of the LIS administration [1, 2].
- The need for the combination of the several access control models in the case of the continuous functioning of subsystems where these principles of the access control are implemented [3 – 5].

Now interest in RBAC is shifted towards the role mining problem [6 – 9] and different “supermodels” [10, 11]. One more direction of scientific activities is development of methods of RBAC configuration for cloud computing [12, 13].

Based on these factors a set of the problems which are important for the effective work of a LIS safety subsystem is selected.

1. The creation and the supporting of the RBAC.

1.1. The deleting the duplicate roles.

1.2. The taxonomic distribution of permissions.

1.3. The optimal file representation of a role hierarchy.

1.4. The assessment of risks of the leakage of permissions.

2. The access control on the basis of the computable cryptographic keys.

3. The combining of the different security models in a computer system.

The efficiency of the solution of these problems significantly depends on the structure of the role hierarchy which is used in the LIS access control policy. The additional properties of the role hierarchy which are required for the solution of the formulated problems are found: tree-like hierarchy; leaf hierarchy; *RP*-reduced hierarchy; transitive-reduced hierarchy (tab. I) [14, 15].

There is a need for development of the conversion methods of the role hierarchy according to the specified characteristics. At the same time changes of an access control subsystem must be transparent for the user: a user is obliged to receive the same permissions before and after the conversions.

II. Local optimization of the RBAC

The graph-based representation of the RBAC model was formalized for more strict determination of the possible conversions of the role hierarchy. It is known that RBAC model is defined by the set of following elements: U, P, R – sets of users, permissions and roles; RP: $R \rightarrow 2^P$, UR: $U \rightarrow 2^R$, RR: $R \rightarrow 2^R$, UP: $U \rightarrow 2^P$ – mappings which define the distribution of permissions between roles, the authorization of users for roles, the authorization of roles at each other, the users permissions.

The directed marked graph which has no directed cycles (the *role graph*) $G = (R, E, RP)$ is the graphical representation of sets P, R and mappings RP, RR. The arc (r_i, r_j) exists in the role graph if and only if $r_j \in RR(r_i)$.

TABLE I. CHARACTERISTICS OF THE ROLE HIERARCHY WHICH ARE NECESSARY FOR THE SOLUTION OF SOME ACCESS CONTROL PROBLEMS

Role hierarchy	tree-like	leaf	*RP*-reduced	transitive-reduced
Number of the problem	(1.4)		(1.1)	(1.3)
	(2)	(1.2)		
	(3)			

RBAC models are named the *equivalent* models if the sets of permissions coincide, the sets of users coincide too, and the user's permissions mappings are isomorphic in these models.

Required conversions of the RBAC model must consist in the following:

- to lead the role hierarchy to the required look;
- to lead to creation of the equivalent RBAC model;
- to make the minimum changes to the main sets and mappings of the RBAC model.

These conversions are named the *local optimization* of the RBAC model. By definition the local optimization comes down to conversions of a role graph.

For further formalization the *RP*-conversions of a role graph are defined. These conversions must lead to creation of the equivalent RBAC model. The conversion of a role graph G to a role graph G^* is *RP-admissible* if the following conditions are satisfied:

- $RP(G) \subseteq RP(G^*)$;
- $\forall$ directed path $\rho(r_i, r_j) \in G$ $\exists$ "conjugate" directed path $\rho(r_u, r_v) \in G^*$: $RP(r_i) = RP(r_u) \wedge RP(r_j) = RP(r_v)$.

The conversion of a role graph G to a role graph G^* is *RP-equivalent* if the following conditions are satisfied:

- $RP(G) = RP(G^*)$;
- the requirement of the existence of the "conjugate" directed path must be fulfilled both for an initial role graph and for a resultant role graph.

It is proved that the conversion F of a role graph G to a role graph G^* is the local optimization of the RBAC model if the following conditions are satisfied:

- G^* meets the selected criterion of optimality;
- F is *RP*-admissible (or *RP*-equivalent) conversion;
- the number of the nodes and/or of the arcs of a role graph did not increase or this increase is minimum.

Four criteria of the local optimization of a role graph are selected: tree-like role graph, leaf role graph, *RP*-reduced role graph and transitive-reduced role graph.

Some propositions and theorems which define and justify the algorithms of the local optimization of the RBAC model are proved [14, 15]. These algorithms lead to creation of the equivalent RBAC model. Four main and four derivative algorithms are received (tab. II).

TABLE II. ALGORITHMS OF THE LOCAL OPTIMIZATION OF THE RBAC MODEL

Algorism	**Conversion**	**Features of the optimal role graph**
Main algorithms		
(I)	*RP*-admissible	single, leaf
(II)	*RP*-equivalent	*RP*-reduced
(III)	*RP*-equivalent	tree-like
(IV)	*RP*-equivalent	transitive-reduced
Derivative algorithms		
(Ia)	*RP*-admissible	leaf
(I+II)	*RP*-admissible	single, taxonomic, leaf, *RP*-reduced
(III+I)	*RP*-admissible	leaf, tree-like
(III+Ia)	*RP*-admissible	single, leaf, tree-like

The complexity of the constructed algorithms is a polynomial in the number of roles and permissions.

The algorithms of the local optimization of the RBAC model were used for creation of the methods of the solution of some access control problems (tab. 1). The results received for problems (1.4) (2) and (3) follow.

III. APPLICATIONS OF THE ALGORITHMS OF THE LOCAL OPTIMIZATION

The assessment problem of risks of the leakage of permissions in the RBAC policy is formalized: probability of leakage of each permissions $p_i \in P$ must be evaluated. The new algorithm of the solution of this problem uses AHP and the role graph of the RBAC policy. The tree-like leaf role hierarchy is necessary for implementation of the offered approach. This type of hierarchy is provided by algorithm (III+I). The advantages of the developed technique are defined and justified:

- The "model" error of AHP which arises in a consequence of inconsistency of opinions of experts is removed.
- The automation of the process of the ordering of the permissions according to the value of the risks of their leakage is possible.

The problem of the access control in the hierarchical system is defined: the set of the system's objects is partially ordered; access control model for this system must be defined [16]. The execution of the following conditions is supposed:

- The object hierarchy is defined by the digraph G; the order G is n.
- Each object O_i (i = 1, …, n) is encrypted by a secret key k_i (an *access key*); a symmetric encryption is used.
- An access to the object O_i is possible if the access key k_i is known and the condition of a sequential access is satisfied. The sequential access consists in the following: if a user wants to get an access to the object O_i, he must have an access to all objects which form the directed path from the hierarchy root to the object O_i.

In the local systems this problem can be solved by a security subsystem. In the distributed system a uniform security subsystem is absent; the access control is reached by means of algorithms of a distribution of cryptographic keys and an encryption. The new method of the distribution of the access keys is offered. This method is based on the principle of the computability of keys [16]. For the implementation of this approach the object hierarchy must be tree-like. Algorithm (III) provides this requirement.

The new method of the combining of RBAC and mandatory access control (MAC) is offered. This approach is based on the search of the "ideal" solution: the access rules must meet requirements of the both models and must not contradict each other. The Cartesian product of the MAC lattice and the “role” lattice which is generated by the role hierarchy is considered. Algorithm (III) is involved for the receiving a “role” lattice.

Further problems (1.4) and (2) will be represented in more detail.

IV. The Assessment of Risks of the Leakage of Permissions

The heuristic assumptions are formulated:

1. The more the number of permissions of the role is, the more probability of the attack on this role is.

2. The more the prevalence of the permission in the role hierarchy is, the more the probability of the leakage of this permission is.

3. The more the distance of the role from the hierarchy root is, the less the probability of the attack on this role is.

According to algorithm (III+I) of a local optimization of the RBAC model we can assume that the role graph is the tree-like leaf digraph *T*.

At the preparatory stage the tree *T* extends to the tree T_p: to each leaf r_l of the tree *T* the new nodes are added; the new node contains one permission from the set of permissions $RP(r_l)$ of the leaf r_l (fig. 1). The tree of the solution of the AHP is the tree T_p.

At the first stage the relative coefficients of all nodes (except a root) of the tree T_p are calculated. The calculation of the coefficients takes place in the direction from the root to the leaf. The nodes $r_{s1}, \ldots, r_{sk}$ which are subordinated to one node r_s from the previous level are considered. In terms of AHP the selected nodes $r_{s1}, \ldots, r_{sk}$ are the alternatives for the criterion r_s. For each subset $\{r_{s1}, \ldots, r_{sk}\}$ the paired comparisons matrix $\mathbf{M}_s$ is built. This matrix is used for the calculation of the relative coefficients of the nodes $r_{s1}, \ldots, r_{sk}$. The matrix element $[\mathbf{M}_s]_{ij}$ which corresponds to pair (r_{si}, r_{sj}) is equated to the relation of the number of permissions of the role r_i to the number of permissions of the role r_j:

$$[\mathbf{M}_s]_{ij} = |RP(r_{si})| \big/ |RP(r_{sj})|.$$

It is proved that these paired comparisons matrixes are ideally coordinated. As a result the relative coefficient w_{si} of the node r_{si} is calculated according to the formula:

$$w_{si} = |RP(r_{si})| \Big/ \sum\nolimits_{j=1}^{k} |RP(r_{sj})|.$$

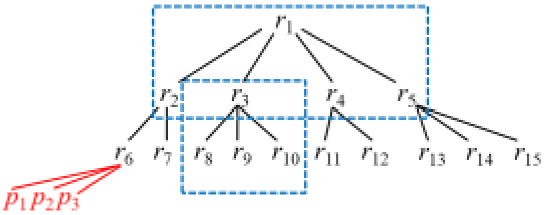


Fig. 1. The fragment of the extended tree T_p

At the second stage the combined coefficients (probabilities or risks of the leakage of permissions) are calculated. The probability of the leakage of the permission *p* is equal to the sum over all directed paths from the root to the leaves which contain this permission. Each item of the sum is the product of the relative coefficients of the nodes which form one of the directed paths.

$$P(p) = \sum_{\rho(\text{root, leaf}):\, RP(\text{leaf}) = \{p\}} \left(\prod_{j:\, r_j \in \rho(\text{root, leaf})} w_j \right).$$

The complexity of the suggested algorithm is a polynomial in the number of roles *n* and permissions *m*: $T = O((n \cdot m)^2)$.

V. The Access Control on the Basis of the Computable Cryptographic Keys.

One secret key k_0 is determined and stored. All access keys are calculated one of another according to an object hierarchy: if an arc (O_i, O_j) is in the hierarchy then the access key k_j of the object O_j is a value of the function *h*, which is infeasible to invert (*h* is one-way function). The function *h* depends on the access key k_i of the object O_i and the properties of the object O_j: $k_j = h(k_i, O_j)$. The algorithm of the formation of computable access keys for a tree-like object hierarchy is defined. This algorithm uses cryptographic hash function as function *h*.

1. The unique identifier id_i is assigned to each object O_i. The general initialized key k_0 is defined.

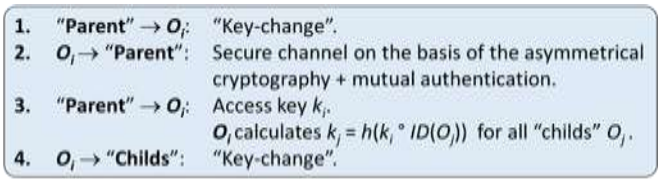
2. The access key k_1 of the object O_1 which is a root of the object hierarchy is calculated by the formula: $k_1 = h(k_0 \circ id_1)$, where $(k_0 \circ id_1)$ is the concatenation of the key and the identifier.

3. For each arc (O_i, O_j) of the object hierarchy the access key k_j of the object O_j is calculated by the formula:

$$k_j = h(k_i \circ id_j).$$

The suggested algorithm is named "hash-based access keys distribution" (HBAKD). It is proved that existence of the tree-like object hierarchy is necessary and sufficient condition of uniqueness of the computation of access keys according to the HBAKD algorithm. The generalized algorithm of the formation of computable access keys for arbitrary object hierarchy which is described by the digraph *G* is defined:

1. The directed tree *T* which is *ID-equivalent* to the digraph *G* is constructed. For this purpose the modification of algorithm (III) of a local optimization of the RBAC model is used.



1. “Parent” → O_i: “Key-change”.
2. O_i → “Parent”: Secure channel on the basis of the asymmetrical cryptography + mutual authentication.
3. “Parent” → O_i: Access key k_i.
   O_i calculates $k_j = h(k_i \circ ID(O_j))$ for all “childs” O_j.
4. O_i → “Childs”: “Key-change”.

Fig. 2. The scheme of "Key-change" protocol

2. According to the HBAKD algorithm the access keys for all nodes of the tree *T* are computed.

3. Secret sharing scheme can be applied to each *ID* class of the tree *T* if this class consists of several elements.

The cryptographic protocols for the practical use of the HBAKD algorithm are offered. The main one is "Key-change" protocol (fig. 2). The possible attacks on these protocols are analyzed. The additional safety measures which are put in protocols are proved (tab. III).

The conditions of the implementation of the discretionary and mandatory principles of the access control on the basis of the HBAKD algorithm are formulated. It is supposed that the set of the objects is partially ordered and the access to any object is possible only sequentially from the "parent" object to the "child" object. It is proved that the implementation of the mandatory and discretionary models of the access control is possible for a tree-like object hierarchy in the case of the execution of the conditions which are listed in table IV (ID_S is the set of the identifiers of the objects which are accessible for the subject *S*).

VI. Conclusions

Some problems which are important for the effective work of an access control policy in a LIS have been discussed. The solution of these problems significantly depended on the structure of the hierarchy of the system's entities. A special attention has been paid to the RBAC model. The general concept of the local optimization of the RBAC model has been formulated. This process consists in the equivalent conversions of a role graph. The equivalent conversions make the minimum changes to the main sets and mappings of the RBAC model and increase the efficiency of the functioning of the system due to the coercion of the role hierarchy to the required look. The tools which can execute the specified conversions of the RBAC model have been developed. On the basis of optimization algorithms the methods of the creation and of the supporting of the access control policy in a LIS have been obtained. The correctness of the offered approaches, methods and algorithms has been confirmed by the rigorous mathematical proofs and by the results of the computing experiments.

TABLE III. The additional safety measures of the “key-change” protocol

Possible attacks	Safety measures
Interception of the message "Key-change"; substitution of the “child”	Step 2 of the protocol
Sending the false message "Key-change"; substitution of the "parent"	

TABLE IV. Conditions of the implementation of the discretionary and mandatory principles of the access control

Access control	Identifiers	It is known to the *S*
Mandatory	Open	k_i
Discretionary	Secret	$k_0 + ID_S$
Mandatory and discretionary	Secret	$k_i + ID_S$

References

[1] Boadu E.O., Armah G.K. Role-Based Access Control (Rbac) Based In Hospital Management. International Refereed Journal of Engineering and Science (IRJES). 2014. Vol. 3, Issue 9. P. 53–67. URL: http://www.irjes.com/Papers/vol3-issue9/H395367.pdf.

[2] Cheung H., Li C., Yu Y., Yang C. Privacy Protection for Role-Based Access Control in Service Oriented Architecture. International Journal of Network Security & Its Applications (IJNSA). 2014. Vol. 6, No. 3. DOI: 10.5121/ijnsa.2014.6301.

[3] Amini M., Arasteh M. A combination of semantic and attribute-based access control model for virtual organizations. International Journal of Information Security. 2015. Vol. 7, Issue 1. P. 27-45. DOI: 10.22042/ISECURE.2015.7.1.4.

[4] Shi S., Xu C. Design and Implementation of a Role-Based Access Control for Categorized Resource in Smart Community Systems. Cloud Computing and Big Data (CCBD). 2016. 7th International Conference. DOI: 10.1109/CCBD.2016.044.

[5] Mallare I.J.G., Pancho-Festin S. Combining Task- and Role-Based Access Control with Multi-Constraints for a Medical Workflow System. IT Convergence and Security (ICITCS). 2013. International Conference. DOI: 10.1109/ICITCS.2013.6717814.

[6] Mitra B., Sural S., Atluri V., Vaidya J. The generalized temporal role mining problem. Journal of Computer Security. 2015. Vol. 23, Issue 1. P. 31–58. URL: https://dl.acm.org/citation.cfm?id=2746190.

[7] Mitra B., Sural S., Vaidya J., Atluri V. A Survey of Role Mining. ACM Computing Surveys (CSUR). 2016. Vol. 48, No. 4. P. 1–37. DOI: 10.1145/2871148.

[8] Huang H., Shang F., Liu J., Du H. Handling least privilege problem and role mining in RBAC. Journal of Combinatorial Optimization. 2014. Vol. 30, Issue 1. P. 63–86. DOI: 10.1007/s10878-013-9633-9.

[9] Zhang W., Chen Y., Gunter C., Liebovitz D., Malin B. Evolving role definitions through permission invocation patterns. Proceedings of the 18th ACM symposium on Access control models and technologies. 2013. P. 37–48. DOI: 10.1145/2462410.2462422.

[10] Liu C., Peng Z., Wu L. Role of Time-Domain Based Access Control Model. Journal of Software Engineering and Applications. 2016. Vol. 9, No. 2. DOI: 10.4236/jsea.2016.92004.

[11] Bogaerts J., Lagaisse L., Joosen W. Idea: Supporting Policy-Based Access Control on Database Systems. In: Caballero J., Bodden E., Athanasopoulos E. (eds) Engineering Secure Software and Systems. ESSoS 2016. Lecture Notes in Computer Science, vol 9639. Springer, Cham. DOI: 10.1007/978-3-319-30806-7_16.

[12] Tang Z., Wei J., Sallam A., Li K., Li R. A New RBAC Based Access Control Model for Cloud Computing. In: Li R., Cao J., Bourgeois J. (eds) Advances in Grid and Pervasive Computing. 2012. Lecture Notes in Computer Science. Vol. 7296. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-30767-6_24.

[13] Li W., Wan H., Ren X., Li S. A Refined RBAC Model for Cloud Computing. Computer and Information Science (ICIS). 2012. IEEE/ACIS 11th International Conference. DOI: 10.1109/ICIS.2012.13.

[14] Bogachenko N.F. Local Optimization of the Role-Based Access Control Policy. CEUR Workshop Proceedings. 2017. Vol. 1965. URL: http://ceur-ws.org/Vol-1965/paper14.pdf.

[15] Belim S., Bogachenko N., Ilushechkin E. An analysis of graphs that represent a role-based security policy hierarchy. Journal of Computer Security. 2015. V. 23, No. 5. P. 641–657. DOI: 10.3233/JCS-150532.

[16] Belim S.V., Bogachenko N.F. Distribution of Cryptographic Keys in Systems with a Hierarchy of Objects. Automatic Control and Computer Sciences. 2016. Vol. 50, No. 8. P. 777–786. DOI: 10.3103/S0146411616080071.