sciendo

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 4
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0044

Access Control Models

Maria Penelova

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,1113 Sofia, Bulgaria
E-mail: i_n_f@abv.bg

***Abstract*:** *Access control is a part of the security of information technologies. Access control regulates the access requests to system resources. The access control logic is formalized in models. Many access control models exist. They vary in their design, components, policies and areas of application. With the developing of information technologies, more complex access control models have been created. This paper is concerned with overview and analysis for a number of access control models. First, an overview of access control models is presented. Second, they are analyzed and compared by a number of parameters: storing the identity of the user, delegation of trust, fine-grained policies, flexibility, object-versioning, scalability, using time in policies, structure, trustworthiness, workflow control, areas of application etc. Some of these parameters describe the access control models, while other parameters are important characteristics and components of these models. The results of the comparative analysis are presented in tables. Prospects of development of new models are specified.*

***Keywords*:** *Access control, authorization, access control model, permission, access control policy.*

1. Introduction

Access control is an important part from the information security technologies. Another term for access control is authorization. Authorization denotes that an access request to software resource is granted or denied, depending on the permissions of the user and the access control rules. The logic for authorization is formalized in access control models. The components of an access control model are: a set of subjects, a set of objects, a set of operations, a set of permissions and a set of policies. A subject is a human being, a computer process, a robot, or a device. An object is a software resource. An operation is a kind of action, for which the subject makes an access request for the object. A permission shows that a subject can access an object through an operation. A policy is a rule that shows if the access request has to be granted or denied.

Many access control models exist. The first of them, Identity-Based Access Control has been published in 1969, in the work of Lampson – an access control matrix [18]. Two popular access control models are based on access control matrix – Access Control Lists (ACLs) and Capabilities.

In 1970, the multilevel method for access control has been published in a security report. It provides extra security to computer systems. In 1973, Bell and LaPadula [1] have formalized the multilevel method to a mathematical model. This allows the properties of the model to be examined and analyzed in detail. In 1976, Harrison, Ruzzo and Ullman have shown that the access control matrix is undecidable [14].

In 1983, Discretionary Access Control (DAC) and Mandatory Access Control (MAC) are introduced [8]. They are very important access control models, which, in combination, ensure the security of computer systems.

Role-Based Access Control (RBAC) family of reference models have been published in 1996. It introduces "role" as part of access control model. The roles express the policy of RBAC. This model is the most popular access control model. RBAC is used for enterprise systems.

Some other models use the role concept of RBAC. They add different kinds of policies, access control parameters and components to the model. This is described and analyzed in the paper.

An important step is the publishing the specification of Attribute-Based Access Control specification by National Institute of Standards and Technology (NIST) of The United States in 2014. In introduces "attribute" as a part of access control model.

The specification of Next Generation Access Control [36, 37] is expected to be developed by NIST, after the concept has already been described [82]. The document published by now is a draft. This model uses attributes, too.

With the developing of information technologies, more complex access control models have been created. They meet the new requirements of Internet of things [95], ubiquitous computing, cloud computing [94], online social networks [97], web services, relational databases, smart collaborative ecosystems [96], artificial intelligence [98], data sharing on smart devices [99], etc.

Nowadays, there are research papers, that are concerned with analysis of access control policies, models and mechanisms [89-92]. Access control mechanisms [3, 84] are enhanced. An existing access control model has been unified in [93]. Authorization problem has been detected [101]. Surveys and reviews of access control models in particular areas of application have been published [102-104].

The mentioned above access control models and other are described and compared in this paper: Context-Based Access Control (CBAC), View-Based Access Control (VBAC), Token-Based Access Control (TokenBAC), Relationship-Based Access Control (ReBAC), Provenance-Based Access Control (PBAC), etc. The models are analyzed and compared by a number of parameters: storing the identity of the user, delegation of trust, fine-grained policies, flexibility, object-versioning, scalability, using time in policies, structure, trustworthiness, workflow control, areas of application, etc.

The rest of this paper is structured as follows: Section 2 “An Overview of Access Control Models” introduces access control models; comparative analysis of access control models is proposed in Section 3. The results are presented in tables. Section 4 presents the prospects of development and conclusions.

2. An overview of access control models

Nowadays, the access control in information technologies is dynamically developing and offers many solutions.

2.1. Identity-based access control

Identity-Based Access Control (IBAC) is the oldest access control model. It is introduced in 1969 by Lampson [18]. IBAC is represented by an access control matrix. The rows of the matrix belong to users, and the columns pertain to the objects. The cell (*i*, *j*) specifies the access rights of the user *i*, which he/she has to the object *j*. An access right can be own, read, write, execute, and etc.

Two access control models are related to IBAC: ACLs and Capabilities.

2.2. ACLs

Access Control Lists (ACLs) are projection of access control matrix by columns. ACL is a list of permissions, which are granted to a user. This approach is applied in file systems. An example for ACLs of a file is [Mary: read; Alex: read, write, execute;]. That means that Mary can only read this file, but Alex can read, write and execute it.

2.3. Capabilities

Capabilities are projection of access control matrix by rows. A Capability list is attached to each subject, which contains the access rights on each object. Capabilities require cryptography to protect authorization data from reading and change. Some of the access control models considered in this paper – ZBAC and TokenBAC, are based on Capabilities approach.

2.4. Harrison, Ruzzo and Ullman access control model

In 1976, Harrison, Ruzzo and Ullman [14] have analyzed the access control matrix of Lampson for decidability. They have shown that their model reaches such a state, that a subject has a privilege that it did not possess before. This means that in general, safety is undecidable in access control matrix and IBAC. This undecidability is passing from IBAC into DAC – another of the access control models being considered in this paper.

2.5. Multilevel method and related mathematical models

An access control method has been published in 1970, in a RAND Corporation report [31]. It has been called multilevel, because of the multiple security levels of the data.

The access is regulated, depending on the clearance level of the user and the classification (security) level of the object.

In 1973, Bell and LaPadula [1] have formalized multilevel method into a mathematical model. Two basic rules are described in that model: the simple security rule and the *-property (star-property). The simple security rule means that a user at a specific clearance level is not allowed to read information above this level. The *-property denotes that a user cannot write information, that is classified below his/her clearance level. The model of Bell and LaPadula ensures confidentiality in a system.

Biba [2] has published in 1977 a mathematical model. In the simple integrity property of that model, a user is allowed to read information that has security level greater than his/her clearance level. The integrity *-property states that a user can write to object, when the security level of the object is lower that the clearance level of the user. It is important to combine the model of Bell-LaPadula and the model of Biba, to ensure both confidentiality and integrity of a software system.

2.6. Mandatory access control

In 1983 Mandatory Access Control (MAC) has been introduced in Trusted Computer System Evaluation Criteria (TCSEC) [8], published by United States Department of Defence. MAC is based on Bell-LaPadula mathematical model. A characteristic of MAC is passing a data flow in one direction through a lattice of security labels [22]. MAC is used with DAC and is applied mainly in military applications. Security labels are assigned to users and objects, in order to express MAC policy. A label that is assigned to user is called a security clearance. A label assigned to object is called a security classification. MAC policy is mandatory and it is not possible for a user to change it. An example for a module, that includes a MAC policy, is Security-Enhanced Linux (SELinux).

2.7. Discretionary access control

Discretionary Access Control (DAC) is introduced in TCSEC [8], together with MAC. A characteristic of DAC is that the owner of an object can pass access permissions for this object on discretionary principle [22]. Very often the owner of an object is its creator. The access to DAC object is regulated depending on the identity of a user. DAC policies have the greatest application due to their flexibility. DAC is not sufficient to ensure that a system is secure, that is why this access control model is introduced together with MAC. DAC is applied in operating systems in combination with other access control models.

2.8. Role-based access control

A family of Role-Based Access Control (RBAC) models has been introduced in 1996 [25]. RBAC is based on Bell LaPadula mathematical model [10]. Characteristic of RBAC is that permissions are assigned to roles, and users are assigned to proper roles [11, 26]. Role is a job function within an organization. For example, the user with job accountant is assigned to role “Accountant” in a software system. The accountant

permissions are assigned to the role “Accountant”. The result of applying RBAC is a simplified management of permissions. The policy of RBAC is expressed via roles.

The family of RBAC models consists of four components. The base model is $RBAC_0$. The advanced model, $RBAC_1$, includes $RBAC_0$, but supports role hierarchies in addition. The advanced model, $RBAC_2$, includes $RBAC_0$, but with added constraints. The consolidated model, $RBAC_3$, includes $RBAC_1$ and $RBAC_2$. The base RBAC model, $RBAC_0$ consists of the set of users, the set of roles and the set of permissions. A user can be a human being, a robot or a computer. A role is a job function in an organization. A permission is an access right. RBAC supports features as flexibility, scalability, workflow control and separation of duties. RBAC is used in enterprise software. This model is the most popular access control model, due to the flexible policy, focused on roles.

Hybrid Access Control (HAC) has been proposed [63] in 2020. This model extends RBAC and implements the dynamic conflict of interest. HAC is applied in secure localization of satellite and vehicles, based on Internet of things.

2.9. Attribute based access control

Attribute Based Access Control (ABAC) specification [15] of National Institute of Standards and Technology (NIST) of The United States has been published in 2014. The name of the model comes from “attributes”, which are characteristics of the subjects and the objects. A subject can be a human being or a device. An object is the requested resource of a software system. Policy in ABAC is a rule, which specifies whether a subject can access an object. The environmental conditions include date and time, and the location of the user. ABAC access control mechanism evaluates the attributes, the environmental conditions and the policies and makes an access decision. ABAC access control mechanism consists of Policy Decision Point and Policy Enforcement Point. Examples for subject attributes are the name, the role and the job within the organization. ABAC allows subjects and objects that do not exist in the system yet to be included in a policy. ABAC is scalable, flexible and fine-grained. ABAC is applied in enterprise software and web services [27, 32].

2.10. Next generation access control

Next Generation Access Control (NGAC) [36] is flexible, scalable and uses different types of policies together. It is manageable, even when technology changes, organization restructures or the amount of data increases. NGAC is suitable for the software of a distributed and interconnected enterprise. NGAC presents a unifying framework, which can support traditional and new kinds of policies for access control together. NGAC is based on ABAC and uses attributes for authorization. In NGAC, there are attributes of a subject, object and a process. NGAC request consists of a process identifier, user identifier, operation and a sequence of operands, which are supported by the operation.

In NGAC, policies reside in the memory of the computer, not in the disk, like in ABAC [37]. NGAC uses the correct policies and attributes to calculate the access decision. Access decision is made by applying a combining algorithm to policies that do not interfere with each other. In NGAC, administrative operations are used for

managing attributes and policies, but policies are enforced by the access control function. ABAC does not recognize administrative operations and manages policies via interface in Policy Administration Point, which is different from the access control interface.

2.11. Organization-based access control

Organization-Based Access Control (OrBAC) has been introduced in 2003 [47]. This access control model has rules that express contextual permissions, prohibitions, obligations or recommendations. Rules in OrBAC are particular for the organization. In OrBAC, the organization is important element. Organization entity consists of subjects, who have agreed to form it. Subjects are users or organizations. Role is a link between subjects and organizations. Objects are entities, which can be files, database records or emails. Action is another entity, like “read” or “write”. View is a set of objects, which have a common property. There are the following relationships Employ, Consider, Permission, and Define between some of the entities in OrBAC. OrBAC is used in organization applications. This model can be combined with Task-Based Access Control and applied in workflow systems [48].

2.12. Task-based access control

Task-Based Access Control (TaskBAC) has been introduced in 1997 [83]. It is designed for “active” or “dynamic” systems, which consider the context of the task completion in the enterprise [43]. TaskBAC is used for workflow management in environments that consist of tasks. Granting, monitoring and revoking of permissions are done automatically and bind with the progression of the tasks, so TaskBAC is a flexible access control model. Task-Role Based Dual System Access Control [44] has the advantages of TaskBAC and RBAC: sequence of tasks and using roles. Another access control model Task-Oriented Multilevel Cooperative Access Control, based on workflow [45], is applied in cloud computing and Internet of things.

2.13. Risk-based access control

Risk is the probability of an incident that may occur and cause damages. Risk-Based Access Control (RiskBAC) has been introduced in 2004 [66]. It is based on risk estimation [38, 39]. Main modules of RiskBAC are risk estimation, access policies and access decision. Risk estimation module fetches the access request of the user. After analysis about risk factors, the module estimates a value of the security risk, corresponding to the access request. This value is compared to access control policies, in order to make a decision whether to grant or deny access. RiskBAC is flexible and is suitable for systems where context needs to be considered. Such systems are called “dynamic” systems. RiskBAC is used in Internet of things [40], collaborative spam detection [42] and cloud computing [41].

2.14. Rule-based access control

Rule-Based Access Control (RuleBAC) has been introduced in 2005 [59]. It is applied in web-based social networks and decentralized systems [56]. A network is

represented by a graph, where users are nodes and edges are the relationships between the users. RuleBAC uses concept of roles as policies [58]. There are access constraints, related to the type, depth and trust level of the relationship with other users. A depth of relationship is the shortest path, corresponding to a relationship between two users. Model-transformation enhances flexibility to RuleBAC [46, 57].

2.15. Trust-based access control

Trust-Based Access Control (TrustBAC) has been formalized in 2012 [64]. It extends RBAC. In TrustBAC, a level of trust is associated with a user [65]. The trust level is automatically reduced if the behavior of the user deviates from the expected, in order to prevent a misuse. TrustBAC is implemented in distributed applications, Web services, peer-to-peer networks, large-scale computing systems, spam detection, online auctions, reputation systems, cloud computing [67, 68], online social networks and ubiquitous computing [69]. TrustBAC is used in e-Business [71], e-Learning [70] and XML databases [72], too. This model is fine-grained, provides scalability for distributed application.

In SECURE Trust Model [66], there are dimensions, called trust-contexts, which are represented by trust-values. The trust is computed by checking, whether the evidence is appropriate to the current trust-context. The evidence consists of surveillances of former cooperation with this subject and warrants from other participants. Trust calculator computes all corresponding to the subject trust-contexts.

2.16. History-based access control

History-Based Access Control (HBAC) has been published in 1999 [87]. HBAC is a mechanism for computing access rights during the execution of a piece of program code [60, 62]. The general concept of this model is remembering the history of computation. Any piece of code has initial rights, called static rights. Current rights are the access rights during execution at each moment. The checking phase is, when an access decision has to be made. Then, the current rights are by default the access rights during the execution. The storage phase is, when the current access rights are represented as variable, in order the programs to read or update this variable. The phase of automatic updates is, when the code is executed. Then the current rights are updated with the intersection of the old current rights and the static rights. The phase of explicit modification occurs, when a piece of executing code calls a special operation that modifies the current rights. This operation can restore the static access rights of the code. The syntax phase is controlling the modification of rights to use programming patterns. There is special syntax when granting access rights and accepting the results from execution of untrusted programming code.

HBAC is applied in Java Virtual Machines, Common Language Runtime and XML documents [61].

2.17. Context-based access control

Context-Based Access Control (CBAC) dates from 2001 [5]. In this model, there are associated properties to users, resource and environment for access control purpose.

CBAC uses constraints to add context-based policies to RBAC. There are three types of context components: physical, virtual and social [28]. The physical components are: geographical location of the device, date and time and the type of the device. The virtual components are digital signature and public key. The social component of the context is the position of an employee. A trust level is a number in the diapason [0, 1], which is associated with every component of the context. A role is assigned to a participant, according to the values of the trust levels of the context components.

A device sends an access request. The request is accepted and the Access Control Service for the permission is called. If the user is authenticated, the rules of access control policies are applied and the role is assigned to the participant. The access permission is granted, depending on the role of the user.

CBAC is applied in ubiquitous computing [6, 7] and Internet of things. CBAC is used for multimedia medical image database systems [30] and Smart Space [28].

2.18. View-based access control

A view is a virtual table that includes data (rows and columns) from one or more database tables. A view can be used in a query like a database table. View-Based Access Control (VBAC) has been introduced in 2001 [88]. It regulates access to views [19]. Access control policy is implemented in two steps in a database. First, the views are created with queries. Second, the access privileges are granted. VBAC uses roles. VBAC provides fine-grained access control and is suitable for relational databases [20, 29].

2.19. Authorization-based access control

AuthoriZation-Based Access Control (ZBAC) has been presented in 2009 [17]. It is similar to capability-based models. Users are authenticated via a service. Authentication in the domain of the user is made before the access request. That authentication generates at least one authorization, which is implied by encrypted credentials and assertions. An authorization is valid for a specific duration of time. The service checks whether the authorization is valid, in order to grant access. In ZBAC, it is possible not to store the identity of the user. Each permission is represented by an explicit authorization. An argument can be passed to the authorization, in order to provide fine-grained access control. ZBAC is created for distributed and service-based systems.

2.20. Relationship-based access control

A social network is a directed graph with multiple types of edges. Nodes represent users, the different types of edges represent the different types of relationships between users. ReBAC model has been published in 2011 by F o n g and S i a h a a n [12] and F o n g [13]. In ReBAC, access control is based on the relationships between the resource owner and the resource requestor in a social network. The access control policies support delegation of trust. ReBAC catches the context of relationships. Characteristics of the model are: tracking of interpersonal relationships between users

and using of their relationships in access control policies. ReBAC is used for online social networks.

2.21. Provenance-based access control

Provenance-Based Access Control (PBAC) has been introduced in 2012 [23]. The features of PBAC are: workflow control, origin-based control and object-versioning. The main components of the model are: artifacts, processes and agents. There are different types of dependencies between two components. The main components and the dependencies generate a directed acyclic graph. In this graph, the nodes are represented by main components and the edges represent the dependencies.

Artifacts capture data objects, and the processes capture functional actions. The agents are users. PBAC uses provenance data, in order to grant or deny access to a resource.

A family of PBAC models has been introduced. $PBAC_B$ is the Base model that includes captured and computable provenance data, object dependencies and a policy. $PBAC_U$ extends $PBAC_B$ by allowing User-declared provenance data. $PBAC_A$ extends $PBAC_B$ by including Acting user dependencies. $PBAC_{PR}$ extends the base model to include provenance-based Policy Retrieval. Combinations of the three extended models can exist.

PBAC is used in cloud technologies [21, 24].

2.22. Attribute-based encryption access control

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) has been introduced in 2007 [79]. CP-ABE model includes five algorithms [75]. The first algorithm, Setup, generates a public key. The second, KeyGen, produces a private key, which is based on the attributes of the subject. The third, Encrypt, generated a ciphertext. That ciphertext can be decrypted only by the user, who has the attributes that satisfy a tree access structure. The fourth algorithm, Decrypt, performs decryption. The fifth algorithm, Delegate, produces a secret key for a set of attributes. CP-ABE is applied in cloud computing [73]. This model is flexible and fine-grained.

Another fine-grained access control solution, based on CP-ABE is [74].

2.23. Token-based access control

Token-Based Access Control (TokenBAC) has been introduced in 2005 [78]. It is similar to capabilities and ACLs. User must have an access token and must show it to the system, in order to get a resource. The differences between TokenBAC and ACLs/Capabilities are: TokenBAC does not store the identity of a user, and the user, not the system, regulates the tokens. Tokens are generated by token manager. They are automatically linked to access request. The system checks whether the application, that requests a resource, has at least one of the stored tokens. In this case, and when there are no associated tokens with input data, the access is granted. A characteristic of TokenBAC is decentralization.

TokenBAC is used in distributed applications, blockchain, ubiquitous computing applications [16], Internet of things [4], cloud computing [9].

2.24. Dynamic and semantic-aware access control

Dynamic and Semantic-Aware Access Control (DSAAC) [33] is an identity-based access control model. It has been published in 2020. DSAAC is developed, assuming workflow process in environment for multiple data centers. By assessing the violations of the sequence of the work process and semantic constraints, the access of the users to the objects is controlled. In DSAAC, the request for object includes attributes and historical behavior request. Via risk assessment at each task from the workflow, the access is denied or the administrators are warned for irregularities. Administrators can examine the access decisions, edit the sequence pattern library and update the module for detecting sequence anomalies. DSAAC is suitable for dynamic access control in environments with multiple resources.

2.25. Lightweight collaborative ciphertext policy attribute role-based encryption

Lightweight Collaborative Ciphertext Policy Attribute Role-Based Encryption (LW-C-CP-ARBE) scheme [34] has been introduced in 2021. LW-C-CP-ARBE is flexible and fine-grained model that provides privacy-aware outsourced data sharing. Due to lightweight proxy re-encryption protocol and privacy-aware policy, it is possible to control read and write access in mobile cloud environment. LW-C-CP-ARBE minimizes data re-encryption and decryption cost. Access control policies are encrypted and thus, they are stored hidden in the cloud.

2.26. Access control model for distributed database systems

A Scalable and Expandable Access Control (SEAC) [35] model has been published in 2020. It is designed for distributed database systems. SEAC is easy for management and provides scalability, better functionality and consistence. The model consists of: objects, users, security dimensions, access levels and permission levels. Security dimensions contain values, that are assigned to users. Permission levels allow to update the security settings of an object. Access levels allow to display or edit an object. Permissions and access levels are calculated automatically, according to the security dimension values, in order to provide more efficient access control.

2.27. Blockchain access control

Blockchain technology consists of linked blocks that cannot be modified [50]. Blockchain blocks are validated from the participants, called miners, in peer-to-peer network. The notion of blockchain appears after the genesis of Bitcoin, which is online cryptocurrency, which manages the transactions in a decentralized peer-to-peer network. Blockchain is decentralized, distributed, irreversible, traceable and tamper-proof technology. Every miner shares the set of linked blocks in blockchain. Blockchain access control approaches [55] are used in Internet of things [53], for creating smart cities, and in healthcare systems [54].

Blockchain access control is presented in [49]. It is based on TokenBAC and ABAC policies and implemented in Bitcoin. The user, who is a resource owner, creates two kinds of tokens in a transaction. The first token passes access rights from

one subject to another. The second token helps to update or revoke the policies, specified by the owner. Policies and attributes are stored in outer system, which reduces the intricacy of blockchain, but causes the following disadvantages: unavailability, mutability, insecurity. The enforcement of policies is not self-executed.

Blockchain smart contracts are used for decentralized, flexible and fine-grained access control for smart buildings [51] and dynamic access control [52]. A smart contract is code that is executed on a blockchain to enforce an agreement between the participants. Each contract represents a unique access. If a transaction is executed successfully, the status of the smart contract is changed. The abbreviation (BACSC) in Table 1 stands for Blockchain Access Control with Smart Contracts.

Blockchain access control is used in health record systems, too [81].

3. Comparative analysis of access control models

The access control models are analyzed and compared by a number of parameters: storing the identity of the user, delegation of trust, flexibility, scalability, fine-grained policies, object-versioning, using time in policies, structure, trustworthiness, workflow control, area of application, and etc. These characteristics are achieved, sometimes, in different ways.

3.1. Storing the identity of the user

Storing the identity of the user is important characteristic of an access control model. TokenBAC and ZBAC do not store the identity of the user. The rest of the models store the identity of the user in the system. The characteristic “Identity” in Table 1 shows whether the model stores the identity of the user in the system.

3.2. Dynamic models

Distributed and workflow management systems require “active” or “dynamic” models for access control. CBAC, VBAC, ReBAC, RiskBAC, TaskBAC, OrBAC, TrustBAC and DSAAC are dynamic access control models. CBAC and VBAC use the current context, and that is why they are dynamic. ReBAC uses the context of the relationship. RiskBAC uses the context of the access request for access control. In TaskBAC, the progression of the executing tasks supports “dynamic” access control. OrBAC can be combined with TaskBAC, which makes OrBAC dynamic. TrustBAC uses trust-context. DSAAC assesses the anomaly in user access requests, and that is why this access control model is dynamic. The other access control models are not dynamic. The characteristic “Dynamic” in Table 1 shows whether the access control model is dynamic.

3.3. Delegation of trust

Delegation of trust shows whether the model passes privileges from one user to another user, based on trusted relationship between the users. Relationships in ReBAC use context, which supports delegation of trust. There is no data for other

models to support delegation of trust. The characteristic “Delegation of Trust” in Table 1 shows whether the model supports delegation of trust.

Table 1. The result or comparative analysis of access control models

Characteristic	Model																								
	IBAC	ACLs	C	DAC	MAC	RBAC	ABAC	NGAC	CBAC	VBAC	ZBAC	REBAC	PBAC	TokenBAC	BACSC	RiskBAC	TaskBAC	ORBAC	RuleBAC	TrustBAC	HBAC	CPABE	DSAAC	SEAC	LWCCPARBE
Identity	+	+	+	+	+	+	+	+	+	+	-	+	+	-	+	+	+	+	+	+	+	+	+	+	+
Dynamic	-	-	-	-	-	-	-	-	+	+	-	+	-	-	-	+	+	+	+	+	-	-	+	-	-
Delegation of Trust	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-
Fine-grained	-	-	-	-	-	-	+	+	-	+	+	-	-	-	+	-	-	-	-	+	-	+	-	-	+
Flexible	-	-	-	+	-	+	+	+	+	+	-	+	-	+	+	+	+	-	+	+	-	-	+	-	-
Object-versioning	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-
Scalability	-	-	-	-	-	+	+	+	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Constraints	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	+	+	+	-	+	-	-	-	-
SoD	-	-	-	-	-	+	-	-	-	-	-	+	+	-	-	-	+	-	-	-	-	-	-	-	-
Time	-	-	-	-	-	-	+	+	+	-	-	+	+	+	-	-	-	-	+	+	+	-	-	-	-
Location	-	-	-	-	-	-	+	-	+	-	-	+	-	+	-	-	-	-	+	-	-	-	-	-	-
Tree-based	-	-	-	-	-	-	+	+	-	-	+	+	-	-	-	-	-	-	-	-	-	+	-	-	+
Trustworthy	-	-	-	-	-	-	-	-	+	-	+	+	+	-	-	-	-	-	-	+	-	-	-	-	-
Workflow control	-	-	-	-	-	+	-	-	-	-	-	-	+	-	-	-	+	+	-	-	-	-	+	-	-
Encryption	-	-	+	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	+	-	-	+
Attributes	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-	+
Roles	-	-	-	-	-	+	-	+	+	-	-	-	-	-	-	+	-	+	+	+	-	-	-	-	+
Tamper-proof	-	-	+	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-
Decentralized	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-
Smart contracts	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-
Tokens	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-
Authorizations	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Access levels	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Permission levels	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Security dimensions	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Rules	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-
Tasks	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	+	-	-
History keeping	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+	-	-	-	-
Relationships	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-
Ciphertexts	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	+
Certificates	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-
Distributed	-	-	-	-	-	-	-	+	+	-	+	+	+	+	+	-	+	-	-	+	-	+	-	+	+
Risk factors	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	+	-	-
Views	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Context	-	-	-	-	-	-	-	-	+	+	-	+	-	+	-	+	+	+	+	+	-	-	+	-	-
Organizations	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-

In Table 1, C stands for capabilities; “+” denotes that the model has a specific characteristic; “-” denotes, that the access control model does not possess specific characteristic.

3.4. Fine-grained policies

Fine-grained policies denote the ability of the policies of the model to ensure more detailed access control check. RBAC is not fine-grained, because it prevents an operation to be executed, but it does not protect specific data. ABAC and NGAC are fine-grained, due to the policies, which evaluates the attributes. TrustBAC is fine-grained, because it computes trust-context. VBAC is fine-grained, due to the access control for the different granularities in the database. ZBAC is fine-grained, because an argument can be passed to the authorization, assigned to each permission. CP-ABE and LW-C-CP-ARBE are fine-grained, due to attributes that describe the private key of the user. Blockchain access control, based on smart contracts is fine-grained, because each smart contract corresponds to a unique access. HBAC is fine-grained, because the application being executed is split down to basic operations [85]. The other access control models are not fine-grained access control models. They are coarse-grained access control models. The characteristic “Fine-grained” in Table 1 shows whether the model has fine-grained policies.

3.5. Flexibility

Flexibility is the ability of the policies of the access control model to adjust to the area of application. For example, discretionary policies of DAC have great application in operating systems, due to their flexibility. The policies of RBAC, RuleBAC and TrustBAC are flexible, too, but they are based on roles. In ABAC, flexibility is achieved by making a dynamic access control decision, which is based on attributes. NGAC can apply different types of policies and that is why it is flexible. TaskBAC is flexible, because access control decisions are made automatically and are bound to the progression of the tasks. The flexibility of RiskBAC, CBAC and DSAAC is due to the context, used in the policies. RuleBAC has enhanced flexiblity, when it is described from metamodels [57]. The policy of blockhain access control with smart contracts is flexible, because access rules and users can be declared as invalid, and it is not necessarily users and resources to be deleted [51]. VBAC is flexible for database security, because uses views, not tables; the policies are flexible, because support access control rules, depending on the context [76]. The flexibility of TokenBAC is due to the context, used in the policies [77]. The characteristic “Flexible” in Table 1 shows whether the model has flexibility.

3.6. Object-versioning

Object-versioning shows the ability of an access control model to create versions of its objects. PBAC supports many historical copies that are versions of one object and therefore object-versioning is a characteristic of that model. The rest of the models do not support object-versioning. The characteristic “Object-versioning” in Table 1 shows whether the access control model supports object-versioning.

3.7. Scalability

Scalability shows whether the model can work with increasing number of users and objects of the software system. RBAC, ABAC, NGAC, VBAC, ZBAC and SEAC

are scalable access control models. RBAC and ABAC are scalable for enterprise systems. NGAC is scalable for distributed enterprise systems. VBAC is scalable for relational databases. ZBAC is scalable for distributed systems and web services. SEAC is scalable for distributed database systems. There is no information for the other models, whether they are scalable. The characteristic “Scalability” is used in Table 1.

3.8. Constraints, different from separation of duty

Constraints ensure safety in access control models. There are two types of constraints: static and dynamic. The check for static constraints is made when the access rights are assigned. Dynamic constraints are applied with the access request during runtime. Separation of duty is the most popular kind of constraint, but there are other constraints, that are specific to access control models. In RuleBAC, there are access constraints, related to the type, depth and trust level of the relationship with other users [56]. In ABAC, the policies require the attributes to be constrained and to have allowable values [15]. There are temporal constraints to the access history elements in HBAC [86]. In OrBAC, constraints are represented by rules that are applied to different relations [47]. TaskBAC has static constraints, such as processing states, trustee-sets, protection states and executor permissions [80]. The dynamic constraints in TaskBAC are: dynamic separation of duty, dynamic separation of roles and coincidence of roles. There is no information for the rest of the access control models to include constraints, which are different from separation of duty. The characteristic “Constraints” in Table 1 shows whether the access control model supports constraints that are different from separation of duty.

3.9. Separation of Duty

Separation of duty manages conflict of interests in distributed applications. There are static separation of duty and dynamic separation of duty. In RBAC, static separation of duty denotes that a user cannot be member of role A and role B, while dynamic separation of duty is, when a user cannot use role A and role B in one session. In ReBAC, well-formed contexts represent dynamic separation of duty [13]. PBAC and TaskBAC support dynamic separation of duty. There is not information for other models to support separation of duty. The abbreviation of characteristic “SoD” in Table 1 stands for separation of duty.

3.10. Time

Time is used in the policies of the following models: ABAC, NGAC, CBAC, TokenBAC, ReBAC, RuleBAC, TrustBAC, HBAC and PBAC. In ABAC, time is included in the environmental conditions. In NGAC, time is a condition [82]. Time is included in the context in CBAC. In TokenBAC, tokens and environmental data, like time, are stored in the system. ReBAC includes time in the context of the relationship. TrustBAC includes time in the trust-context. HBAC and PBAC keep history, therefore they use time in their policies. There is no information for the other

access control models to use time. The characteristic “Time” in Table 1 shows whether the access control model uses time in its policies.

3.11. Location

Some models use location as access control parameter. Such access control models are: ABAC, CBAC, TokenBAC, ReBAC and RuleBAC. In ABAC, location is included in the environmental conditions. In CBAC, location is included in physical components of the context. In TokenBAC, tokens and environmental data, like location, are stored in the system. ReBAC includes location in the context of the relationship. Location may be present in condition part of a rule in RuleBAC. There is no information for the other models to use location. The characteristic “Location” in Table 1 shows whether the access control model uses location in its policies.

3.12. Tree-based structure

Access control models, such as ABAC and NGAC, that use attributes, have tree-based structure. Models, like PBAC and ReBAC, that represent graphs, have tree-based structure. CP-ABE and LW-C-CP-ARBE use access trees, therefore they have tree-based structure. In RuleBAC, the online social networks are represented by graphs. The other access control models do not have tree-based structure. The characteristic “Tree-based” shows whether the access control model has tree-based structure.

3.13. Trustworthy

Trustworthiness denotes that the data is passed from trusted user, or trusted object or trusted context. In CBAC, there are trust level values, which are calculated according to the context. In ZBAC trust relationships are encoded in authorizations. ReBAC supports also and trust delegation. PBAC ensures trustworthy provenance data. TrustBAC is based on trust, therefore this model is trustworthy. In RuleBAC, there are trust levels that are assigned to relationships between the users [56]. In HBAC, the access control is managed by establishing trust relationships [85]. There is no data for the other models to be trustworthy. There is a characteristic “Trustworthy” in Table 1.

3.14. Workflow control

Workflow control shows whether the model allows tracking of the work process via its policies. RBAC and PBAC support workflow control. TaskBAC, OrBAC and DSAAC are designed to track the work process. There is no information for the other access control models to support workflow control. There is a characteristic “Workflow control” in Table 1.

3.15. Using encryption

Capabilities, CP-ABE, LW-C-CP-ARBE, TokenBAC and Blockchain access control with smart contracts use encryption to protect the stored access control data. The

other access control models do not use encryption. There is a characteristic "Encryption" in Table 1, which shows whether the access control model uses encryption.

3.16. Using attributes

Attributes are characteristics of the subjects and the objects. ABAC, NGAC, DSAAC, CP-ABE and LW-C-CP-ARBE use attributes. ABAC supports subject attributes and object attributes. In NGAC, there are subject and object attributes, that are mutable. Attributes are mutable, when they change after different access requests, while immutable attributes are updated only by the administrator. In ABAC, attributes are immutable. DSAAC supports characteristic attributes of the access request, that include subject attributes, object attributes and other attributes of the request. CP-ABE and LW-C-CP-ARBE describe the private key of a user with attributes. In these models, the leaves in the access tree of the ciphertext describe these attributes. If the attributes of the user correspond to the leaves in the access tree, the user can decrypt a ciphertext. There is a characteristic "Attributes" in Table 1, which shows whether the access control model uses attributes.

3.17. Using roles

Roles are used as policies in some models. RBAC, CBAC, TrustBAC, RuleBAC, OrBAC,VBAC and LW-C-CP-ARBE use roles. The other access control models do not use roles. There is a characteristic "Roles" in Table 1, which shows whether the access control model uses roles.

3.18. Tamper-proof

Capabilities use tamper-proof mechanisms, so the user cannot change his/her capabilities list. Blockchain is a tamper-proof technology. Tamper-proof means immutability, which is result of authorizing and validating the new blocks by all the participants in the network. Any hacker attack for change can be easily recognized and prevented. The access control models, TokenBAC and Blockchain access control with smart contracts, applied in blockchain, inherit the tamper-proof property. The other access control models are not used in tamper-poof technologies. There is a characteristic "Tamper-proof" in Table 1.

3.19. Decentralized

Some models are used in decentralized systems. Decentralization denotes, that all the subjects in the access control model perform access control processes. The opposite of decentralization is centralization, when the system or network administrator is responsible for authorization. Blockchain is a decentralized technology, so both access control models, TokenBAC and Blockchain access control with smart contracts, that are applied in blockchain are used in decentralized systems. The rest of the access control models are not used in decentralized systems. There is a characteristic "Decentralized" in Table 1.

3.20. Smart contracts

A smart contract is code that is executed on a blockchain, in order to support the agreement between the participants in a transaction. A smart contract is used for encoding a random state-transition function, too. A unique address is assigned to a contract. A transaction is sent to this address for execution, by the user. When a request for transaction execution is received, a callback function is executed. The state of smart contracts changes, only if the transaction has successfully finished. Smart contracts are introduced in Blockchain Access Control with Smart Contracts (BACSC) [51]. In other access control models smart contracts do not present. There is a characteristic “Smart contracts” in Table 1.

3.21. Tokens

Users possess tokens [16], which are shown to the system, in order an access decision to be made. The users track the tokens, but the system does not. Using tokens does not require users to be identified. Assigning time and location information to the objects makes the access control scheme effective using tokens. Objects are associated with a set of secret tokens. Tokens and environmental data, such as time and location are stored in the system. Subjects, whose access requests are granted, have provided copies of the corresponding tokens, before that. Tokens are used in decentralized systems. TokenBAC uses tokens, but the other access control models do not use tokens. There is a characteristic “Tokens” in Table 1.

3.22. Authorizations

ZBAC uses authorizations that are presented with the access request [17]. An authorization represents every permission that is exercised. An argument can be passed to an authorization, enabling fine-grained access control. The rest of the models do not use authorizations. There is a characteristic “Authorizations” in Table 1.

3.23. Access levels

Access levels are used in SEAC. They specify the type of access to an object. There can be read, write or no access to an object, according to access level value [35]. In the rest of models, there are no access levels. There is a characteristic “Access levels” in Table 1.

3.24. Permission levels

An object may be queried, according to permission levels of that object. If a permission level has value “Allowed”, the access request of the user for an object may be proceeded. If a permission level is “Not Allowed”, the access request is denied and the object is not displayed [35]. SEAC uses permission levels, while the other models do not use permission levels. There is a characteristic “Permission levels” in Table 1.

3.25. Security dimensions

A security dimension expresses a characteristic of the users. According to that characteristic, there are some values, which describe different users. A security dimension consists of all these values [35]. For example, the security dimension, called “Job position” may have values as “Employee”, “Manager” and “Administrator”. SEAC model uses security dimensions, while the rest of the models do not use security dimensions. There is a characteristic “Security dimensions” in Table 1.

3.26. Rules

A rule assigns or denies a permission to a particular subject. A rule consists of a condition part and a decision part [57]. Decision part can be “accept” or “deny”. Condition part includes data, such as source address, destination address, time, etc. RuleBAC model uses rules. The rest of the models do not use rules. There is a characteristic “Rules” in Table 1.

3.27. Tasks

TaskBAC uses tasks for access control purposes. In TaskBAC, permissions are permanently monitored. They can be made active or inactive, depending on the context, which is the current state of a task. The progression of tasks determines the access control decision [80]. The rest of the models do not use tasks. There is a characteristic “Tasks” in Table 1.

3.28. History keeping

Both HBAC and PBAC keep history for regulating the access requests. The main difference between HBAC and PBAC is, that HBAC remembers history about the behavior of the subjects, while PBAC stores provenance data about objects [23]. The rest of the models do not keep historical data. There is a characteristic “History keeping” in Table 1.

3.29. Relationships

In a model that uses relationships, the access control decisions depend on the relationship between the owner of the resource and the user, who makes the resource request, in a social network. Relationships use context [13]. ReBAC uses relationships for access control. In RuleBAC [56], there are relationships between users in online social networks. The rest of the models do not use relationships. There is a characteristic “Relationships” in Table 1.

3.30. Ciphertexts

Ciphertexts are used in distributed systems. A ciphertext is computed by encryption of an access tree. That access tree consists of descriptive attributes that identify the private keys of the users. A user can decrypt a ciphertext with a specified private key if the attributes from that key correspond to the nodes of the access tree. In a

ciphertext is formulated the access policy of the model [79]. CP-ABE and LW-C-CP-ARBE use ciphertexts for access control. In comparison, LW-C-CP-ARBE reduces the computation cost of CP-ABE. LW-C-CP-ARBE supports read and write access to a resource, while CP-ABE provides read only access for a user, that is not data owner. The rest of the models do not use ciphertexts. There is a characteristic "Ciphertexts" in Table 1.

3.31. Certificates

In RuleBAC, certificates are created and signed between users, if a direct relationship exists between them with a specific trust level [56]. There is no information for the other access control models to use certificates. There is a characteristic "Certificates" in Table 1.

3.32. Distributed

Some access models are designed for distributed systems. NGAC is created for distributed enterprise. CBAC is designed for distributed systems, like Smart Space. ZBAC is designed for distributed systems and web services. ReBAC is applied in online social networks and supports distributed access control. PBAC is created for distributed systems, too. TokenBAC and Blockchain access control with smart contracts are applied in blockchain, which is a distributed technology. TaskBAC, TrustBAC, CP-ABE, LW-C-CP-ARBE are suitable for distributed computing, too. SEAC is designed for distributed databases. The other models are not designed for distributed systems. There is a characteristic "Distributed" in Table 1.

3.33. Risk factors

RiskBAC regulates the access requests, on the basis of risk factors evaluation. In DSAAC, the access is denied or the administrators are warned for irregularities, via risk assessment at each task from the workflow. The rest of access control models do not use risk factors evaluation. There is a characteristic "Risk factors" in Table 1.

3.34. Views

A view is a virtual table that includes data (rows and columns) from one or more database tables. A view can be used in a query like a database table. VBAC uses views. In VBAC, the access control policy is implemented in two steps in a database. First, the views are created with queries. Second, the access privileges are granted. The rest of access control models do not use views. There is a characteristic "Views" in Table 1.

3.35. Context

CBAC, VBAC, TokenBAC, RiskBAC and DSAAC use context-based policies. The relationships have context in ReBAC. There are well-formed contexts [13]. Permissions are contextual in OrBAC. In TrustBAC, there is trust-context. The context is linked with the progression of the tasks in TaskBAC. There is no

information for other access control models to use context. There is a characteristic "Context" in Table 1.

3.36. Organizations

Organizations are included as entities in OrBAC. The rest of the models do not use organizations. There is a characteristic "Organizations" in Table 1.

The results are presented in Table 1.

Access control models can be researched and analyzed for area of application (Table 2). Each access control model is designed for specific technologies.

Table 2. The application areas of access control models

Model	Area of application
ABAC	Enterprise software, cloud computing, web services
ACLs	Operating systems
Capabilities	Operating systems
DAC	Operating systems
IBAC	Operating systems
MAC	Military applications, Mail servers and operating systems
RBAC	Enterprise software, information systems
CBAC	Firewalls, ubiquitous computing and Internet of things
VBAC	Relational databases
TokenBAC	Distributed applications, blockchain, ubiquitous computing, Internet of things and cloud computing
ReBAC	Online social networks
PBAC	Cloud technologies
ZBAC	Distributed and service-based systems
BACSC	Blockchain technologies
RiskBAC	Internet of things, collaborative spam detecting and cloud technologies
TaskBAC	Enterprise software [83], cloud technologies and Internet of things
OrBAC	Organization applications and workflow systems
RuleBAC	Web-based social networks and decentralized systems
TrustBAC	Distributed applications, web services, peer-to-peer networks, large-scale computing systems, spam detection, online auctions, reputation systems, cloud computing, online social networks and ubiquitous computing, e-Business, e-Learning, XML databases
HBAC	Java Virtual Machines, Common Language Runtime, XML documents, Autonomic Grid Services, Mobile Code [87]
CP-ABE	Cloud computing
DSAAC	For environments with multiple resources
SEAC	Distributed database systems
LW-C-CP-ARBE	Mobile cloud environment
NGAC	Distributed and interconnected enterprise

4. Prospects of development and conclusions

This paper presents a number of access control models and the areas, where they are applied. IBAC, ACLs and DAC are used in operating systems. MAC is applied for military applications. ZBAC is designed for distributed and service-based systems. PBAC has an application in cloud technologies. ReBAC is used for online social networks. TokenBAC is related to distributed applications, blockchain, ubiquitous computing applications, Internet of things and cloud computing. CBAC is used for protection of traffic through firewalls, ubiquitous computing and Internet of things. VBAC is designed for relational databases. RBAC and ABAC are applied in enterprise software.

The access control models considered have been analyzed and compared by a number of parameters: storing the identity of the user, delegation of trust, fine-grained policies, flexibility, object-versioning, scalability, using time in policies, structure, trustworthiness, workflow control, areas of application, and etc.

Prospects of development are expressed in creating hybrid access control models and new access control solutions for the following areas: cloud computing, Internet of things, blockchain, mobile cloud environment, smart collaborative ecosystems, artificial intelligence, data sharing on smart devices and distributed databases.

This analysis is made, in order to develop a new access control model, which is in a separate article. The new model have been designed for enterprise software and information systems.

References

1. Bell, D., L. LaPadula. Secure Computer Systems: Mathematical Foundations and Model. Bedford, MA, The Mitre Corporation, 1973.
2. Biba, K. Integrity Considerations for Secure Computer Sytems. – In: Technical Report ESD {TR {76-372, The MITRE Corporation, HQ Electronic Systems Division, Hanscom AFB, MA, April 1977.
3. Schlegal, M. Poster: Shielding AppSPEAR – Enhancing Memory Safety for Trusted Application-level Security Policy Enforcement. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 99-101. **https://doi.org/10.1145/3450569.3464396**
4. Claeys, T., F. Rousseau, B. Tourancheau. Securing Complex IoT Platforms with Token Based Access Control and Authenticated Key Establishment. – In: Proc. of International Workshop on Secure Internet of Things (SIOT), September 2017, Oslo, Norway. Hal-01596135, 2017, pp. 1-9. DOI: 10.1109/SIoT.2017.00006.
5. Covington, M., W. Long, S. Srinivasan, A. K. Dey, M. Ahamad, G. D. Abowd. Securing Context-Aware Applications Using Environment Roles. – In: Proc. of 6th ACM Symposium on Access Control Models and Technologies '01, Chantilly, Virginia, USA, May 2001, pp. 10-20. **https://doi.org/10.1145/373256.373258**
6. Corradi, A., R. Montanari, D. Tibaldi. Context-Based Access Control Management in Ubiquitous Environments. – In: Proc. of 3rd IEEE International Symposium on Network Computing and Applications (NCA'04), Cambridge, MA, USA, 30 August-1 September 2004, pp. 253-260. DOI: 10.1109/NCA.2004.1347784.

7. C o r r a d i, A., R. M o n t a n a r i, D. T i b a l d i. Context-Based Access Control for Ubiquitous Service Provisioning. – In: Proc. of 28th International Computer Software and Applications Conference (COMPSAC'04), Design and Assessment of Trustworthy Software-Based Systems, 27-30 September 2004, Hong Kong, China, Proceedings. IEEE Computer Society, September 2004, pp. 444-451. DOI:10.1109/CMPSAC.2004.1342877.
8. Trusted Computer System Evaluation Criteria (TCSEC). Department of Defence, USA, 5200.28-STD, 1983.
9. E t h e l b e r t, O., F. M o g h a d d a m, P. W i e d e r, R. Y a h y a p o u r. A JSON Token-Based Authentication and Access Management Schema for Cloud SaaS Applications. – In: Proc. of 5th International Conference on Future Internet of Things and Cloud (FiCloud'17), IEEE, 2017, pp. 47-53. DOI: 10.1109/FiCloud.2017.29.
10. F e r r a i o l o, D., D. K u h n, R. C h a n d r a m o u l i. Role-Based Access Control. Second Edition. Artech House, 2007.
11. F e r r a i o l o, D., R. S a n d h u, S. G a v r i l a, D. K u h n, R. C h a n d r a m o u l i. Proposed NIST Standard for Role-Based Access Control. – ACM Transactions on Information and System Security, Vol. **4**, August 2001, No 3, pp. 224-274.
12. F o n g, P., I. S i a h a a n. Relationship-Based Access Control Policies and Their Policy Languages. UC CPSC Technical Report 2011-990-02, January 2011, pp. 51-60. **https://doi.org/10.1145/1998441.1998450**
13. F o n g, P. Relationship-Based Access Control: Protection Model and Policy Language. – In: Proc. of 1st ACM Conference on Data and Application Security and Privacy, CODASPY'11, San Antonio, Texas, USA, 21-23 February 2011, pp. 191-202. **https://doi.org/10.1145/1943513.1943539**
14. H a r r i s o n, M., W. R u z z o, J. U l l m a n. Protection in Operating Systems. – CACM, Vol. **19**, August 1976, No 8, pp. 461-471.
15. H u, V., D. F e r r a i o l o, R. K u h n, A. S c h n i t z e r, K. S a n d l i n, R. M i l l e r, S. K a r e n. Guide to Attribute-Based Access Control (ABAC) Definitions and Considerations. – In: NIST Special Publication 800-162, SIN'13, 2014.
16. I a c h e l l o, G., G. A b o w d. A Token-Based Access Control Mechanism for Automated Capture and Access Systems in Ubiquitous Computing. – In: GVU Technical Report; GIT-GVU-05-06, 2005. Last access April 2021. **https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.536.197&rep=rep1&type=pdf**
17. K a r p, A., H. H a u r y, M. D a v i s. From ABAC to ZBAC: The Evolution of Access Control Models. – HP Laboratories, Technical Report HPL-2009-30, 2009. Last access July 2020. **http://www.hpl.hp.com/techreports/2009/HPL-2009-30.pdf**
18. L a m p s o n, B. Dynamic Protection Structures. – In: AFIPS Conference Proceedings, Vol. **35**, 1969, pp. 27-38.
19. L i a n g, Y. Study on View-Based Security Model for Database [J]. – In: Proc. of Sun Yatsen University Forum, 2005, (03), pp. 134-137.
20. L i n, J., Y. F a n g, B. C h e n, L. W a n. View-Based Access Control Mechanism for Spatial Database. – In: Proc. of International Conference on Earth Observation Data Processing and Analysis (ICEODPA), Vol. **7285**, 728535, Wuhan, China 2008. **https://doi.org/10.1117/12.815569**
21. N g u y e n, D., J. P a r k, R. S a n d h u. Adopting Provenance-Based Access Control in OpenStack Cloud IaaS. – In: Proc. of 8th International Conference Network and System Security NSS, 2014, pp 15-27.
22. O s b o r n, S., R. S a n d h u, Q. M u n a w e r. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. – ACM Transactions on Information and System Security, Vol. **3**, May 2000, No 2, pp. 85-106.
23. P a r k, J., D. D g u y e n, R. S a n d h u. A Provenance-Based Access Control Model. – In: Proc. of 10th Annual International Conference on Privacy, Security and Trust (PST'12), IEEE, 2012, pp. 137-144.
24. R a m a n e, M., B. V a s u d e v a n, S. A l l a p h a n. A Provenance-Policy Based Access Control Model for Data Usage Validation in Cloud. – Cryptography and Security, Vol. **3**, October 2014, No 5, pp. 1-9. **https://arxiv.org/ftp/arxiv/papers/1411/1411.1933.pdf**

25. S a n d h u, R., E. C o y n e, H. F e i n s t e i n, C. Y o u m a n. Role-Based Access Control Models. – In: IEEE Computer, 1996, pp. 38-47.
26. S a n d h u, R., D. F e r r a i o l o, R. K u h n. The NIST Model for Role-Based Access Control: Towards a Unified Standard. – In: Proc. of 5th ACM Workshop on Role-Based Access Control, ACM, 2000, pp. 47-63.
27. S c h l ä g e r, C., M. S o j e r, B. M u s c h a l l, G. P e r n u l. Attribute-Based Authentication and Authorization Infrastructures for e-Commerce Providers. – In: Proc. of EC-Web, 2006, pp 132-141.
https://doi.org/10.1007/11823865_14
28. S m i r n o v, A., A. K a s h e v n i k, N. S h i l o v, N. T e s l y a. Context-Based Access Control Model for Smart Space. – In: Proc. of 5th International Conference on Cyber Conflict K. Podins, J. Stinissen, M. Maybaum, Eds. NATO CCD COE Publications, Tallinn, 2013, pp. 1-15.
29. R o s e n t h a l, A., E. S c i o r e. Content-Based and View-Based Access Control. – In: H. C. A. Van Tilborg, S. Jajodia, Eds. Encyclopedia of Cryptography and Security, Boston, MA, Springer, 2011, pp. 11-59.
https://doi.org/10.1007/978-1-4419-5906-5_695
30. T z e l e p i, S., D. K o u k o p o u l o s, G. P a n g a l o s. A Flexible Content and Context-Based Access Control Model for Multimedia Medical Image Database Systems. – In: Proc. of 4th Workshop on Multimedia & Security: New Challenges, MM&Sec 2001, Ottawa, Ontario, Canada, 5 October 2001, pp. 52-55.
https://doi.org/10.1145/1232454.1232473
31. W a r e, W. Security Controls for Computer Systems (U). – In: Report of Defense Science Board Task Force on Computer Security, Santa Monica, CA: The RAND Corporation, February 1970.
32. Y a n, E., J. T o n g. Attributed Based Access Control (ABAC) for Web Services. – In: Proc. of IEEE International Conference on Web Services, ICWS 2005, Washington, DC, USA, IEEE Computer Society, 2005, pp. 561-569.
33. C h e n, A., G. L u, H. X i n g, Y. X i e, S. Y u a n. Dynamic and Semantic-Aware Access-Control Model for Privacy Preservation in Multiple Data Center Environments. – International Journal of Distributed Sensor Networks 2020, Vol. **16**, 2020, No 5.
https://doi.org/10.1177/1550147720921778
34. F u g k e a w, S. A Fine-Grained and Lightweight Data Access Control Model for Mobile Cloud Computing. – In: IEEE Access, Vol. **9**, 2021, pp. 836-848. DOI: 10.1109/ACCESS.2020.3046869.
35. G u c l u, M., C. B a k i r, V. H a k k o y m a z. A New Scalable and Expandable Access Control Model for Distributed Database Systems in Data Security. – In: Hindawi Scientific Programming. Vol. **2020**. 2020, Article ID 8875069. 10 p.
https://doi.org/10.1155/2020/8875069
36. InterNational Committee for Information Technology Standards (INCITS). Information Technology – Next Generation Access Control. Implementation Requirements, Protocols and API Definitions. USA, 2017.
37. Information Technology Laboratory National Institute of Standards and Technology, US. Department of Commerce. Exploring the Next Generation of Access Control Methodologies. ITL Bulletin for November 2016.
38. D i e p, N., L. H u n g, Y. Z h u n g, S. L e e, Y. L e e, H. L e e. Enforcing Access Control Using Risk Assessment. – In: Proc. of 4th European Conference on Universal Multiservice Networks, Toulouse, France, 14-16 February 2007, pp. 419-424.
39. A t l a m, H., M. A z a d, M. A l a s s a f i, A. A l s h d a d i, A. A l e n e z i. Risk-Based Access Control Model: A Systematic Literature Review. – Future Internet, Vol. **12**, 2020, No 6, 103.
https://doi.org/10.3390/fi12060103
40. A t l a m, H., A. A l e n e z i, R. W a l t e r s, G. W i l l s, J. D a n i e l. Developing an Adaptive Risk-Based Access Control Model for the Internet of Things. – In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 655-661.
DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.103.

41. A l u v a l u, R., K. C h e n n a m, A. J a b b a r, S. A h a m e d. Risk Aware Access Control Model for Trust Based Collaborative Organizations in Cloud. – International Journal of Engineering and Technology (UAE), 2018, pp. 49-52. DOI: 10.14419/ijetv7i4.6.20235.
42. D i m m o c k, N., J. B a c o n, D. I n g r a m, K. M o o d y. Risk Models for Trust-Based Access Control (TBAC). – In: P. Herrmann, V. Issarny, S. Shiu, Eds. Trust Management. iTrust 2005. Lecture Notes in Computer Science. Vol. **3477**. 2005, Berlin, Heidelberg, Springer, pp. 364-371.
https://doi.org/10.1007/11429760_25
43. T h o m a s, R., R. S a n d h u. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. – In: T. Y. Lin, S. Qian, Eds. Database Security XI. IFIP Advances in Information and Communication Technology. Boston, MA, Springer, 1998, pp. 166-181.
https://doi.org/10.1007/978-0-387-35285-5_10
44. Z h a n g, C., Y. H u, G. Z h a n g. Task-Role Based Dual System Access Control Model. – International Journal of Computer Science and Network Security IJCSNS, Vol. **6**, July 2006, No 7B. Last access April 2021.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.4090&rep=rep1&type=pdf
45. D o n g, J., H. Z h u, C. S o n g, Q. L i, R. X i a o. Task-Oriented Multilevel Cooperative Access Control Scheme for Environment with Virtualization and IoT. – In: Hindawi Wireless Communications and Mobile Computing. Vol. **2018**. 2018, Article ID 5938152. 11 p.
https://doi.org/10.1155/2018/5938152
46. A f o n i n, S. Performance Evaluation of a Rule-Based Access Control Framework. – In: Proc. of 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'16), Opatija, Croatia, 2016, pp. 1414-1418. DOI: 10.1109/MIPRO.2016.7522361.
47. K a l a m, A., R. B a i d a, P. B a l b i a n i, S. B e n f e r h a t, F. C u p p e n s, Y. D e s w a r t e, A. M i e g e, C. S a u r e l, G. T r o u e s s i n. Organization Based Access Control. – In: Proc. of 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03), Lake Como, Italy, 2003, pp. 120-131. DOI: 10.1109/POLICY.2003.1206966.
48. W a n g, B., S. Z h a n g. An Organization and Task Based Access Control Model for Workflow System. – In: K. C. C. Chang et al., Eds. Advances in Web and Network Technologies, and Information Management. APWeb 2007, WAIM 2007. Lecture Notes in Computer Science. Vol. **4537**. Berlin, Heidelberg, Springer, 2007, pp. 485-490.
https://doi.org/10.1007/978-3-540-72909-9_51
49. M a e s a, D., P. M o r i, L. R i c c i. Blockchain Based Access Control. – In: L. Chen, H. Reiser, Eds. Distributed Applications and Interoperable Systems (DAIS'17). Lecture Notes in Computer Science. Vol. **10320**. Springer, Cham., 2017, pp. 206-220.
https://doi.org/10.1007/978-3-319-59665-5_15
50. A b d i, I., F. E a s s a, K. J a m b i, K. A l m a r h a b i. A. A L-G h a m d i. Blockchain Platforms and Access Control Classification for IoT Systems. – In: Symmetry, Vol. **12**, 2020, 1663.
https://doi.org/10.3390/sym12101663
51. B i n d r a, L., K. E n g, O. A r d a k a n i a n, E. S t r o u l i a. Flexible, Decentralized Access Control for Smart Buildings with Smart Contracts. 2021. Last accessed Mart 2021.
https://arxiv.org/pdf/2010.08176v1.pdf
52. G u p t a, R., V. S h u k l a, S. R a o, S. A n w a r, P. S h a r m a, R. B a t h l a. Enhancing Privacy through "Smart Contract" Using Blockchain-Based Dynamic Access Control. – In: Proc. of International Conference on Computation, Automation and Knowledge Management (ICCAKM'20), Dubai, United Arab Emirates, 2020, pp. 338-343. DOI: 10.1109/ICCAKM46823.2020.9051521.
53. D r a m é-M a i g n é, S., M. L a u r e n t, L. C a s t i l l o, H. G a n e m. Augmented Chain of Ownership: Configuring IoT Devices with the Help of the Blockchain. – In: Proc. of 14th EAI Int. Conf. Secur. Privacy Commun. Netw. (SECURECOMM'18), Seattle, WA, USA, Springer, Jun 2018, pp. 1-16.

54. T h w i n, T., S. V a s u p o n g a y y a. Blockchain-Based Access Control Model to Preserve Privacy for Personal Health Record Systems. – In: Hindawi Security and Communication Networks. Vol. **2019**. Article ID 8315614. 15 p.
https://doi.org/10.1155/2019/8315614
55. D a r a, A., A. L o n e b, A. B a b a c, R. N a a z b, F. W u c. Blockchain Driven Access Control Mechanisms, Models and Frameworks: A Systematic Literature Review. Last access Mart 2021.
https://eprint.iacr.org/2020/1379.pdf, 2021
56. C a r m i n a t i, B., E. F e r r a r i, A. P e r e g o. Rule-Based Access Control for Social Networks. – In: R. Meersman, Z. Tari, P. Herrero, Eds. On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. OTM 2006. Lecture Notes in Computer Science. Vol. **4278**. Berlin, Heidelberg, Springer, 2006, pp. 1734-1744.
https://doi.org/10.1007/11915072_80
57. M a r t í n e z, S., J. G a r c í a, J. C a b o t. Runtime Support for Rule-Based Access-Control Evaluation through Model-Transformation. – In: Proc. of 2016 ACM SIGPLAN International Conference on Software Language Engineering (SLE'16), October 2016, pp. 57-69.
https://doi.org/10.1145/2997364.2997375
58. P a n e n d e, M., Y. P r a y u d i, I. R i a d i. Comparison of Attribute Based Access Control (ABAC) Model and Rule Based Access (RBAC) to Digital Evidence Storage (DES). – International Journal of Cyber-Security and Digital Forensics, 2018, pp. 275-282
59. L i, H., X. Z h a n g, H. W u, Y. Q u. Design and Application of Rule Based Access Control Policies. 2005. Last access Mart 2021.
https://www.csee.umbc.edu/csee/research/swpw/papers/zhang.pdf
60. A b a d i, M., C. F o u r n e t. Access Control Based on Execution History. – In: Proc. of NDSS Symposium 2003, 2003. Last access Mart 2021.
https://www.ndss-symposium.org/wp-content/uploads/2017/09/Access-Control-Based-on-Execution-History-Martin-Abadi.pdf
61. R ö d e r, P., O. T a f r e s c h i, C. E c k e r t. History-Based Access Control and Information Flow Control for Structured Documents. – In: Proc. of 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS'07), March 2007, pp. 386-388.
https://doi.org/10.1145/1229285.1229336
62. B a n e r j e e, A., D. N a u m a n n. History-Based Access Control and Secure Information Flow. – In: G. Barthe, L. Burdy, M. Huisman, JL. Lanet, T. Muntean, Eds. Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. CASSIS 2004. Lecture Notes in Computer Science. Vol. **3362**. Berlin, Heidelberg, Springer, 2005, pp. 27-48.
https://doi.org/10.1007/978-3-540-30569-9_2
63. A f t a b, M., Y. M u n i r, A. O l u w a s a n m i, Z. Q i n, M. A z i z, Z a k r i a, N. S o n, V. T r a n. A Hybrid Access Control Model with Dynamic COI for Secure Localization of Satellite and IoT-Based Vehicles. – IEEE Access, Vol. **8**, 2020, pp. 24196-24208. DOI: 10.1109/ACCESS.2020.2969715.
64. X i a o n i n g, M. Formal Description of Trust-Based Access Control. – In: Proc. of 2012 International Conference on Medical Physics and Biomedical Engineering, Physics Procedia, Vol. **33**, 2012, pp. 555-560. ISSN 1875-3892.
https://doi.org/10.1016/j.phpro.2012.05.103
65. B h a t t i, R., E. B e r t i n o, A. G h a f o o r. A Trust-Based Context-Aware Access Control Model for Web-Services. – In: Proc. of IEEE International Conference on Web Services, 2004., San Diego, CA, USA, 2004, pp. 184-191. DOI: 10.1109/ICWS.2004.1314738.
66. D i m m o c k, N., A. B e l o k o s z t o l s z k i, D. E y e r s, J. B a c o n, K. M o o d y. Using Trust and Risk in Role-Based Access Control Policies. – In: Proc. of 9th ACM Symposium on Access Control Models and Technologies (SACMAT'04), June 2004, pp. 156-162.
https://doi.org/10.1145/990036.990062
67. J u n, S. A Trust-Game-Based Access Control Model for Cloud Service. – In: Hindawi Mobile Information Systems. Vol. **2020**. 2020, Article ID 4651205. 14 p.
https://doi.org/10.1155/2020/4651205

68. S u n, P. Research on Cloud Computing Service Based on Trust Access Control. – International Journal of Engineering Business Management, Vol. **12**, 2020, pp. 1-13. Last access April 2021. **https://journals.sagepub.com/doi/pdf/10.1177/1847979019897444**
69. F u, B., D. O'S u l l i v a n. User Centric Trust-Based Access Control Management for Ubiquitous Computing Environments. – In: NOMS Workshops 2008 – IEEE Network Operations and Management Symposium Workshops, Salvador, Brazil, 2008, pp. 265-274. DOI: 10.1109/NOMSW.2007.43.
70. W a n g, S., Q. L i u. Trust-Based Access Control in Virtual Learning Community. – In: Integration and Innovation Orient to e-Society, Vol. **2**. IFIP International Federation for Information Processing. Vol. **252**. Springer-Verlag, US, 2007, pp. 514-520. ISBN 978-0-387-75493-2.
71. D a n i l e s c u, M. Modeling Access Control and User Actions Using Trust-Based Access Control Policies. – Journal of Social Sciences, Vol. **3**, 2020, No 3, pp. 72-84. ISSN 2587-3490. ESSN 2587-3504.
72. A s m a w i, A., L. A f f e n d e y, N. U d z i r, R. M a h m o d. XTRUST: A Severity-Aware Trust-Based Access Control for Enhancing Security Level of XML Database from Insider Threats. – PalArch's Journal of Archaeology of Egypt/Egyptology, Vol. **18**, 2021, No 3, pp. 444-450. **https://archives.palarch.nl/index.php/jae/article/view/5604**
73. S h y n u, P., K. S i n g h. A Comprehensive Survey and Analysis on Access Control Schemes in Cloud Environment. – Cybernetics and Information Technologies, Vol. **16**, 2016, No 1, pp. 19-37.
74. T u, S., S. N i u, M. L i. An Efficient Access Control Scheme for Cloud Environment. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 3, pp. 77-90.
75. A l s h e h r i, S., R. R a j. Secure Access Control for Health Information Sharing Systems. – In: Proc. of 2013 IEEE International Conference on Healthcare Informatics, Philadelphia, PA, 2013, pp. 277-286. DOI: 10.1109/ICHI.2013.40.
76. G a b i l l o n, A., L. L e t o u z e y. A View-Based Access Control Model for SPARQL. – In: 4th International Conference on Network and System Security (NSS'10), September 2010, Melbourne, Australia, pp.105-112. DOI: 10.1109/NSS.2010.35ff. ffhal-01020253f.
77. G a n, G., E. C h e n, Z. Z h o u, Y. Z h u. Token-Based Access Control. – IEEE Access, Vol. **8**, 2020, pp. 54189-54199. DOI: 10.1109/ACCESS.2020.2979746.
78. Ia c h e l l o, G., G. A b o w d. A Token-Based Access Control Mechanism for Automated Capture and Access Systems in Ubiquitous Computing, 2005. Last access April 2021. **https://smartech.gatech.edu/bitstream/handle/1853/4482/05-06.pdf?sequence=1&isAllowed=y**
79. B e t h e n c o u r t, J., A. S a h a i, B. W a t e r s. Ciphertext-Policy Attribute-Based Encryption. – In: 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 2007, pp. 321-334. DOI: 10.1109/SP.2007.11.
80. T h o m a s, R., R. S a n d h u. Chapter 10. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. – Database Security XI, 1998, Published by Chapman & Hall, pp. 166-181.
81. T h w i n, T., S. V a s u p o n g a y y a. Blockchain-Based Access Control Model to Preserve Privacy for Personal Health Record Systems. – In: Security and Communication Networks, January 2019. **https://doi.org/10.1155/2019/8315614**
82. Sa n d h u, R., J. P a r k. Usage Control: A Vision for Next Generation Access Control. – In: MMM-ACNS 2003, Berlin, Heidelberg, Springer-Verlag, LNCS 2776, 2003, pp. 17-31.
83. T h o m a s, R., R. S a n d h u. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. – In: Proc. of IFIP WG11.3 Workshop on Database Security, Lake Tahoe, California, 11-13 August 1997. DOI: 10.1007/978-0-387-35285-5_10 Source: DBLP. **http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=174015A71B4347DEF3193901D5353958?doi=10.1.1.54.6227&rep=rep1&type=pdf, last access April 2021**
84. H e u t e l b e c k, D., M. B a u r, M. K l u b a. In-Memory Policy Indexing for Policy Retrieval Points in Attribute-Based Access Control. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 59-70. **https://doi.org/10.1145/3450569.3463562**

85. K o s h u t a n s k i, H., F. M a r t i n e l l i, P. M o r i, A. V a c c a r e l l i. H. Fine-Grained and History-Based Access Control with Trust Management for Autonomic Grid Services. – In: Proc. of International Conference on Autonomic and Autonomous Systems (ICAS'06), Silicon Valley, CA, USA, 2006, pp. 34-34. DOI: 10.1109/ICAS.2006.25.
86. R a v a r i, A., J. J a f a r i a n, M. A m i n i, R. J a l i l i. GTHBAC: A Generalized Temporal History Based Access Control Model. – Telecommun Syst, Vol. **45**, 2010, pp. 111-125. DOI 10.1007/s11235-009-9239-9.
87. E d j l a l i, G., A. A c h a r y a, V. C h a u d h a r y. History-Based Access Control for Mobile Code. – In: J. Vitek, C. D. Jensen, Eds. Secure Internet Programming. Lecture Notes in Computer Science. Vol **1603**. Springer, Berlin, Heidelberg, 1999, pp. 413-431.
https://doi.org/10.1007/3-540-48749-2_19
88. B r o s e, G. A View-Based Access Control Model for CORBA. – In: J. Vitek, C. D. Jensen, Eds. Secure Internet Programming. Lecture Notes in Computer Science. Vol. **1603**. Springer, Berlin, Heidelberg, pp. 237-252,
https://doi.org/10.1007/3-540-48749-2_10
89. M a s o u m z a d e h, A., P. N a r e n d r a n, P. I y e r. Towards a Theory for Semantics and Expressiveness Analysis of Rule-Based Access Control Models. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 33-43.
https://doi.org/10.1145/3450569.3463569
90. C h e n, E., V. D u b r o v e n s k i, D. X u. Mutation Analysis of NGAC Policies. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 71-82.
https://doi.org/10.1145/3450569.3463563
91. J a c o b, F., L. B e c k e r, J. G r a s h ö f e r, H. H a r t e n s t e i n. Matrix Decomposition: Analysis of an Access Control Approach on Transaction-Based DAGs without Finality. – In: Proc. of 25th ACM Symposium on Access Control Models and Technologies (SACMAT'20), June 2020, pp. 81-92.
https://doi.org/10.1145/3381991.3395399
92. E n c k, W. Analysis of Access Control Enforcement in Android. – In: Proc. of 25th ACM Symposium on Access Control Models and Technologies (SACMAT'20), June 2020, pp. 117-118.
https://doi.org/10.1145/3381991.3395396
93. R a d h i k a, B., N. N a r e n d r a K u m a r, R. S h y a m a s u n d a r. Towards Unifying RBAC with Information Flow Control. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 45-54.
https://doi.org/10.1145/3450569.3463570
94. A l-L a i l, M. Poster: Towards Cloud-Based Software for Incorporating Time and Location into Access Control Decisions. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 55-57.
https://doi.org/10.1145/3450569.3464395
95. X u, S., J. N i n g, J. M a, X. H u a n g, H. P a n g, R. D e n g. Expressive Bilateral Access Control for Internet-of-Things in Cloud-Fog Computing. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 143-154.
https://doi.org/10.1145/3450569.3463561
96. G u p t a, M., R. S a n d h u. Towards Activity-Centric Access Control for Smart Collaborative Ecosystems. – In: Proc of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 155-164.
https://doi.org/10.1145/3450569.3463559
97. U l u s o y, O., P. Y o l u m. Norm-Based Access Control. – In: Proc. of 25th ACM Symposium on Access Control Models and Technologies (SACMAT'20), June 2020, pp. 35-46.
https://doi.org/10.1145/3381991.3395601
98. R o s a, M., F. C e r b o, R. L o z o y a. Declarative Access Control for Aggregations of Multiple Ownership Data. – In: Proc. of 25th ACM Symposium on Access Control Models and Technologies (SACMAT'20), June 2020, pp. 59-70.
https://doi.org/10.1145/3381991.3395609

99. M o m e n, N., S. B o c k, L. F r i t c s c h. Accept - Maybe - Decline: Introducing Partial Consent for the Permission-Based Access Control Model of Android. – In: Proc. of 25th ACM Symposium on Access Control Models and Technologies (SACMAT'20), June 2020, pp. 71-80. **https://doi.org/10.1145/3381991.3395603**
100. H e u t e l b e c k, D. Demo: Attribute-Stream-Based Access Control (ASBAC) with the Streaming Attribute Policy Language (SAPL). – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 95-97. **https://doi.org/10.1145/3450569.3464397**
101. C r a m p t o n, J., E. E i b e n, G. G u t i n, D. K a r a p e t y a n, D. M a j u m d a r. Valued Authorization Policy Existence Problem. – In: Proc. of 26th ACM Symposium on Access Control Models and Technologies (SACMAT'21), June 2021, pp. 83-94. **https://doi.org/10.1145/3450569.3463571**
102. G a z z a r a t a, G, B. B l o b e l. Access Control Models – A Systematic Review. – Studies in Health Technology and Informatics, Vol. **261**, 2019, pp. 246-252. DOI:10.3233/978-1-61499-975-1-246.
103. Z h a n g, Y., A. M e m a r i a n i, N. B i d i k a r. A Review on Blockchain-Based Access Control Models in IoT Applications. – In: 16th IEEE International Conference on Control & Automation (ICCA'20), 2020, pp. 671-676. DOI: 10.1109/ICCA51439.2020.9264499.
104. A l n e f a i e, S., S. A l s h e h r i, A. C h e r i f. A Survey on Access Control in IoT: Models, Architectures and Research Opportunities. – International Journal of Security and Networks (IJSN), Vol. **16**, 2021, No 1, pp. 60-76.

Received: 03.01.2021; Second Version: 02.04.2021; Third Version: 08.08.2021; Accepted: 17.09.2021